

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«Харьковский политехнический институт»

Л.П.Галайко

**ИСПОЛЬЗОВАНИЕ ЭВМ
В ЭЛЕКТРОТЕХНИЧЕСКИХ
РАСЧЕТАХ**

Тексты лекций
для студентов электротехнических специальностей

Утверждено
редакционно-издательским
советом университета,
протокол №2 от 21.06.2007

Харьков НТУ „ХПИ” 2008

УДК 004.9:621.313

ББК 32.973

Г15

Рецензенти:

А.И. Яковлев, докт. техн. наук, проф.

Национальный аэрокосмический университет «ХАИ».

Т.Е. Василец, канд. техн. наук, доц.

Украинская инженерно- педагогическая академия.

А.Г. Сосков, докт. техн. наук, проф., ХНАГХ

Наведено 15 лекцій з описом чисельних методів розрахунку, фізичних основ задач з електромеханіки, які підлягають розрахунку, лістингів програм на алгоритмічній мові Паскаль, змісту лабораторних робіт та контрольних запитань. Також наведено приклад застосування програми SimuLink з пакету Matlab для вирішення однієї з задач.

Призначено для студентів електромашинобудівного факультету.

Іл. 29. Табл. 22. Бібліогр. 11 назв.

Галайко Л.П.

Использование ЭВМ в электротехнических расчетах. Тексты лекций для студентов электротехнических специальностей / Л.П. Галайко.— Х.: НТУ «ХПИ», 2008.— 140 с.— На рус. яз.

ISBN 978-966-593-599-5

Приведено 15 лекцій з описанням чисельних методів розрахунку, фізичних основ задач з області електромеханіки, в яких використовуються ці методи, лістинги програм на алгоритмічному мові Паскаль, змісту лабораторних робіт та контрольних запитань. Також розглянуто приклад застосування програми SimuLink з пакету Matlab для рішення однієї з задач.

Призначено для студентів електромашинобудівного факультету.

Іл. 29. Табл. 22. Бібліогр. 11 назв.

ББК 32.973

ISBN 978-966-593-599-5

© Л.П. Галайко, 2008

ПРЕДИСЛОВИЕ

Успехи в развитии полупроводниковой техники способствовали созданию электрических машин нового типа, конструкция, принцип действия и методы расчета которых существенно отличаются от таковых для традиционных электрических машин. Существенная нелинейность параметров и несинусоидальность всех переменных этих машин требуют применения широкого спектра численных методов расчета для проектирования. Современный специалист в области электромеханики должен уметь реализовать численные методы на современных ПЭВМ при разработке программ для проектирования электрических машин нетрадиционных конструкций.

В данном издании приведено краткое описание некоторых численных методов, рассмотрены конкретные задачи из области электромеханики, в которых используются эти методы, приведены примеры программ на языке Паскаль и примеры применения пакета SimuLink из пакета программ Matlab для этих методов и задач.

Издание составлено на основе материалов лекций и лабораторных работ, которые автор в течение ряда лет проводила для студентов 2 курса специальности „Электрические машины и аппараты”. Часть материалов лекций входит в дистанционный курс, разработанный автором.

Цель изучения данного курса лекций заключается в следующем:

1) усвоение численных методов расчета, которые находят применение при разработке новых электрических машин нетрадиционных конструкций;

2) усвоение физики явлений в конкретных задачах электромеханики, в которых используются эти численные методы;

3) совершенствование навыков программирования на языке Паскаль и составления моделей для пакета SimuLink.

Настоящее издание состоит из 15 лекций. По каждой теме приведены описание текста лекции, содержание лабораторной работы по теме, контрольные вопросы.

Материал лекций может быть использован, кроме исходного курса „Использование ЭВМ в электротехнических расчетах”, также в других курсах для направления „Электромеханика”, например, „САПР электрических машин”, „Специальные электромеханические преобразователи”, „Электрические машины” и другие.

ЛЕКЦИЯ 1

КУСОЧНО-ЛИНЕЙНАЯ ИНТЕРПОЛЯЦИЯ ФУНКЦИЙ ОДНОЙ ПЕРЕМЕННОЙ С ПОСТОЯННЫМ ШАГОМ

Применение интерполирования в задачах электромеханики.

Термины и определения.

Вывод расчетных выражений.

Создание алгоритма и функции на языке Паскаль для интерполяции и экстраполяции.

Создание программы для определения точности интерполирования с графической иллюстрацией.

1.1. Применение интерполирования в задачах электромеханики

Проектирование и исследование электрических машин производится по методикам, которые можно рассматривать как математическую модель машины. Модель включает аналитические выражения в виде формул и информацию, представленную в форме таблиц. Возникает задача аппроксимации табличной информации аналитической зависимостью $\varphi = f(x)$, которую можно использовать в последующих расчетах.

Существует 2 возможности аппроксимации табличных данных.

- Аппроксимирующая функция $\varphi(x)$ должна пройти через все точки таблицы. Такой способ аппроксимации называется *интерполяцией* {в широком смысле слова}.

- Аппроксимирующая функция $\varphi(x)$ должна сглаживать (усреднять) данные таблицы. Такой способ аппроксимации называется *регрессией*, или *сглаживанием*.

Примеры зависимостей, для которых применяется *интерполяция*:

- кривая намагничивания: зависимость индукции от напряженности для данной марки стали $B = f(H)$;

- зависимость удельных потерь в стали от индукции для данной марки стали $P_{cm} = f(B)$.

Примеры зависимостей, для которых применяется *регрессия*:

- Зависимость удельных потерь в стали от индукции и марки стали $P_{cm} = f(B, \text{марка стали})$;
- Зависимость коэффициента полезного действия от мощности и частоты вращения $\eta = f(P, n)$.

РЕЗЮМЕ

Интерполирование и регрессия широко применяются при разработке программ для проектирования и анализа режимов работы электрических машин.

1.2. Термины и определения

Интерполяция (в узком смысле слова) – нахождение приближенного значения функции f для значения аргумента x , расположенного между узлами таблицы для функции, заданной в форме таблицы (смотри табл. 1.1).

Таблица 1.1 – Значения аргумента и функции в узлах таблицы

x_0	x_1	x_2	—	x_i		x_n
f_0	f_1	f_2	—	f_i		f_n

Число узлов таблицы равно $(n + 1)$.

Узел – точка в расчетной области с заданными значениями аргумента и функции.

Шаг таблицы – разность значений соседних аргументов $h_i = x_i - x_{i-1}$.

В общем случае шаг может быть постоянным $h_i = \text{const}$ или переменным $h_i = \text{var}$.

Экстраполяция – нахождение приближенного значения функции f для значений аргумента x , расположенных за пределами таблицы $x < x_0$ или $x > x_n$.

При относительно небольших интервалах изменения аргумента для интерполяции применяются полиномы общие для всего интервала изменения аргумента (классический полином, полином Ньютона, полином Лагранжа) [3]. При значительных интервалах изменения аргумента

применяется кусочная интерполяция, при которой число полиномов равно числу участков между узлами, т.е. n . Находят применение:

кусочно-линейная интерполяция – степень полиномов равна 1;

квадратичная интерполяция – степень полиномов равна 2;

кубическая интерполяция – степень полиномов равна 3.

Кубические полиномы называются *сплайнами*.

РЕЗЮМЕ

Кусочно-линейная интерполяция и интерполяция сплайнами широко применяются в программах для проектирования и в программах для численного расчета магнитных полей электрических машин. Их применение требует грамотного владения терминами.

1.3. Вывод расчетных выражений

Получим формулу для определения приближенного значения функции на i -м участке таблицы с помощью кусочно-линейной интерполяции с постоянным шагом с помощью рис. 1.1. Отрезок $AC = \varphi(t)$ – приближенное значение функции. Отрезок AB – точное значение функции $f(t)$. При выводе формулы учитываем исходные данные для кусочно-линейной интерполяции с постоянным шагом:

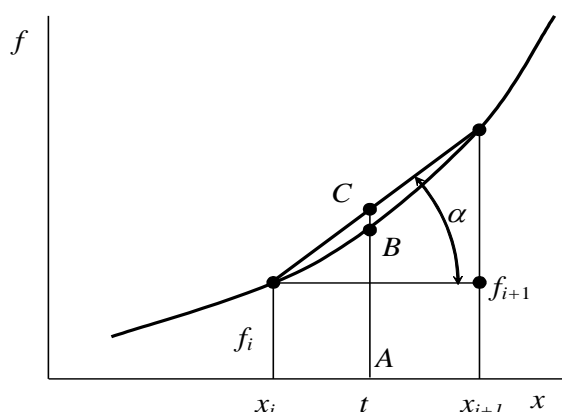


Рисунок 1.1. – К выводу формулы

- 1) шаг h ;
- 2) начальное значение аргумента в таблице x_0 ;
- 3) значения функции в $n + 1$ узлах таблицы;
- 4) число узлов таблицы $n + 1$;
- 5) значение аргумента $x = t$, для которого необходимо найти приближенное значение функции.

$$\varphi(t) = f_i + \Delta_x \operatorname{tg} \alpha; \quad \Delta_x = t - x_i; \quad x_i = x_0 + i h; \quad \operatorname{tg} \alpha = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} = \frac{f_{i+1} - f_i}{h};$$

Окончательная **формула для интерполяции**

$$\varphi(t) = f_i + (t - i \cdot h - x_0) \cdot (f_{i+1} - f_i) / h; \quad x_0 < t < x_n \quad (1.1)$$

Запишем формулы для приближенного определения функции за пределами таблицы. Принимаем наклон аппроксимирующей прямой равным наклону на соседнем участке (при $t < x_0$ – на первом участке, при $t > x_n$ – на последнем участке).

Формулы экстраполяции

$$t < x_0 \quad \operatorname{tg} \alpha = (f_1 - f_0) / h; \quad \Delta_x = t - x_0; \quad \varphi(t) = f_0 + (t - x_0) \cdot (f_1 - f_0) / h; \quad (1.2)$$

$$t > x_n \quad \operatorname{tg} \alpha = (f_n - f_{n-1}) / h; \quad \Delta_x = t - x_0 - n \cdot h; \quad \varphi(t) = f_n + (t - x_0 - n \cdot h) (f_n - f_{n-1}) / h. \quad (1.3)$$

1.4. Создание алгоритма и функции на языке Паскаль для интерполяции и экстраполяции

Если значение аргумента t , для которого необходимо найти приближенное значение функции, находится в пределах таблицы, достаточно использовать только интерполяцию. *Алгоритм интерполяции* состоит из двух этапов:

- 1) определение номера участка, на котором находится значение аргумента t , с помощью формулы $i = \text{целое}((t - x_0) / h)$;
- 2) определение приближенного значения функции на участке по формуле интерполяции (1.1).

Этот алгоритм реализуется в программе в виде функции. *Функция интерполяции* на языке Паскаль имеет вид:

FUNCTION FI(*t:real*):**real**;

Var

i:integer; {определяет номер участка, на котором находится $x=t$ }

BEGIN

i:=trunc(($t-x_0$)/ h); { функция определяет целую часть числа }

FI:=f[i]+(t-x0-i*h)*(f[i+1]-f[i])/h;

END;

Если значение аргумента t может выйти за пределы таблицы,

необходимо использовать *алгоритм интерполяции и экстраполяции*:

- 1) если $t < x_0$ используем формулу экстраполяции (1.2);
- 2) если $t > x_0 + n \cdot h$, используем формулу экстраполяции (1.3);
- 3) иначе используем формулу интерполяции (1.1).

Этот алгоритм реализуется в виде следующей функции на языке Паскаль:

FUNCTION FI(**t:real**):**real**;

var

i:integer;

BEGIN

If $t < x_0$ **then** {экстраполяция}

FI:= $f[0] + (t - x_0) * (f[1] - f[0]) / h$ {формула (1.2)}

Else—

If $t > x_0 + n * h$ **then** {экстраполяция}

FI:= $f[n] + (t - h * n - x_0) * (f[n] - f[n-1]) / h$ {формула (1.3)}

Else {интерполяция}

Begin

$i := \text{trunc}((t - x_0) / h);$

FI:= $f[i] + (t - i * h - x_0) * (f[i+1] - f[i]) / h$; { формула(1.1)}

End;

END;

ДОМАШНЕЕ ЗАДАНИЕ

Записать два – три варианта алгоритма и функции (из 5 возможных) с другой последовательностью формул (1.1), (1.2), (1.3).

*Мы на многое не отваживаемся не потому,
что оно трудно, оно трудно именно потому,
что мы на него не отваживаемся.*

Сенека Луций Анни (6 до н.э.-65 н.э.)

1.5. Создание программы для определения точности интерполирования с графической иллюстрацией

Для определения точности интерполирования необходимо дополнительно к исходной таблице располагать более подробной таблицей с числом

узлов примерно в 10 раз больше. В программе использовалась таблица для кривой намагничивания электротехнической стали марки 3414 (табл. 1.2). Значения функции в узлах этой таблицы удобно хранить во внешнем файле. Данные для исходной таблицы приведены в описательной части головного модуля. Здесь массив f – массив напряженностей.

Таблица 1.2 – Кривая намагничивания электротехнической стали марки 3412

В, Тл	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
	H, А/м									
0.5	65	67	69	71	73	75	77	79	81	83
0.6	85	87	89	91	93	95	97	99	101	103
0.7	105	107	109	111	113	115	117	119	121	123
0.8	125	127	129	131	133	135	137	139	141	143
0.9	145	148	151	154	157	160	163	166	169	172
1.0	176	181	187	193	200	207	214	222	230	240
1.1	250	260	270	280	290	300	310	320	330	340
1.2	351	362	373	384	395	406	417	428	439	450
1.3	461	476	492	508	524	540	556	572	588	604
1.4	620	648	676	704	732	760	788	816	844	872
1.5	900	935	970	1005	1040	1075	1110	1145	1180	1215
1.6	1250	1375	1500	1625	1750	1875	2000	2125	2250	2375

Точность интерполирования будем определять двумя способами:

1) путем расчета относительной погрешности интерполяции в ряде точек между узлами таблицы и за ее пределами по формуле

$$\varepsilon = |(f(t) - \varphi(t)) / f(t)|. \quad (1.4)$$

Значения $f(t)$ находят из внешней таблицы.

2) путем визуального сравнения графиков функций кусочно-линейной интерполяции, построенных по данным исходной таблицы с небольшим числом узлов и по данным таблицы из внешнего файла.

Для первого способа создаем процедуру *ERROR*, для второго – процедуру *VISIOCON*. Кроме того, для графических построений создаем процедуру *GRAPHIC*. Далее приводим полную программу с комментариями.

Program CONTROL;

{Описательная часть}

Uses

Crt, Graph; *{вызываются модули работы с экраном и графикой}*

Const

h:Real=0.1; n:Integer=10;*{шаг и число узлов для исходной таблицы }*

f:Array[0..10] Of

Real=(85,105,120,145,176,250,352,461,620,900,1250);

{ значения функции в узлах исходной таблицы }

x0:Real=0.6;*{начальное значение аргумента в исходной таблице}.*

Procedure ERROR;

Const

xt:Array[0..11] Of Real=(0.55,0.65,0.75,0.85,0.95,1.05,1.15,
1.25,1.35,1.45,1.55,1.65);

{значения аргументов между узлами и за пределами таблицы}

ft:Array[0..11] Of Real=(76,95,115,135,165,200,300,408,540,760,
1075,1875);*{значения функции для аргументов массива xt}*

n1=11;

Var

i:Integer;

fx,e:Real;*{приближенное значение функции и погрешность для аргументов заданных массивом xt}*

Function FI(t:Real):Real;*{функция интерполяции- экстраполяции,вставляем текст из предыдущего раздела 1.4}*

Begin*{головной модуль процедуры ERROR}*

For i:=0 **To** n1 **Do**

Begin

fx:=fi(xt[i]);e:=abs((fx-ft[i])/ft[i]);

```

        writeln('xt=',xt[i]:5:2,' e=',e:10:5);
    End;
End;
Procedure VISIOCON;
Const
    hv=0.01;nv=100;{шаг и число узлов для внешней таблицы}
    name:string='          '{путь к внешнему файлу с таблицей}
Var
    xmax,xmin,ymax,ymin:Real;{данные для построения графика}
    x:Array[0..10] Of Real;{данные для построения графика}
    xv,fv:Array[0..100] Of Real;{данные для построения графика}
    infile:text;{переменная файла}
    i:Integer;
Procedure GRAPHIC;
Var
    i,gd,gm:Integer;
    mx,my:Real;{масштабы по оси x и y}
Begin
    gd:=detect;
    initgraph(gd,gm,' ');{инициализация графического режима,
драйвер- в одной папке с программой}
    mx:=(getmaxx-40)/(xmax-xmin);
    my:=(getmaxy-40)/(ymax-ymin);
    setcolor(red);
    LINE(20,460,620,460);{ось абсцисс}
    LINE(20,460,20,20);{ось ординат}
    setcolor(lightblue);
For i:=1 To n Do
    LINE(20+round(mx*x[i-1]),getmaxy-20-round(my*f[i-1]),20+
    round(mx*x[i]), getmaxy-20-round(my*f[i]));
    {график по исходной таблице}{round- функция округления до
наибольшего целого}

```

```

        setcolor(yellow);
        For i:=1 To nv Do
            LINE(20+round(mx*xv[i-1]),getmaxy-20-round(my*fv[i-1]),
                20+round(mx*xv[i]),getmaxy-20-round(my*fv[i]));
            {график по внешней таблице}
        End;
Begin{исполнительная часть процедуры VISIOCON}
    For i:=0 To 10 Do x[i]:=x0+i*h;{определяются значения коор-
динат
        For i:=0 To 100 Do xv[i]:=x0+i*hv; для построения графиков}
        Assign(infile,name);{устанавливается соответствие между
внешним файлом и переменной файла}
        Reset(infile);{файл открывается для чтения}
        For i:=0 To nv Do Read(infile,fv[i]);{данные считываются в
массив}

        close(infile);{файл закрывается}
        xmin:=0;ymin:=0; xmax:=1.7; ymax:=1500;
        GRAPHIC;{ обращение к процедуре}
        Repeat until keypressed;
        Closegraph;
    End;
    {Главной модуль}
Begin
    ClrScr;
    ERROR;{обращение к процедуре}
    Readln;
    VISIOCON;{обращение к процедуре}
End.

```

1.6. Содержание лабораторной работы

Основные цели лабораторной работы:

- 1) приобрести умение выбирать шаг для получения заданной точности интерполирования;
- 2) ознакомиться с характером кривых намагничивания для разных марок стали;
- 3) совершенствовать навыки программирования при составлении программ для оценки точности интерполирования.

Для достижения 1 цели рекомендуется сначала провести расчеты по описанной программе для стали марки 3414 для шага $h = 0.1$, а затем для значений шагов $h = 0.05, 0.15, 0.2$. Предварительно необходимо обновить значения элементов массивов f , xt , ft , а также значения констант h , n , $n1$ с помощью подробной табл. 1.2. Рекомендуется значения элементов для массива xt выбирать посередине частичных интервалов. Результаты расчета для каждого значения шага необходимо свести в табл. 1.3.

Таблица 1.3 – Значения погрешностей ε для шага $h =$.

№	1	2				i						$n1$
xt_i												
ε												

Проанализировать результаты расчета, отметить значения xt_i , при которых имеет место максимальная погрешность.

Для достижения 2-й цели необходимо повторить расчеты для кривых намагничивания других сталей (одна-две марки), например для стали марки 2212 (данные взять у преподавателя). Сравнить результаты расчета.

Для достижения 3-й цели рекомендуется составить процедуру для автоматического определения значений функции в узлах и между узлами для массивов f и ft при изменении шага с помощью массива fv .

Контрольные вопросы

1. Дайте определение понятию „интерполяция” в широком и узком смысле слова.
2. Дайте определение понятиям „узел” и „шаг таблицы”.
3. Перечислите исходные данные, необходимые для интерполяции с постоянным шагом.
4. Дайте определение понятию „экстраполяция”.
5. Что означает функция *trunc* в языке Паскаль?
6. Назначение функций *Assign* и *Reset* в языке Паскаль.
7. Как определяется наклон аппроксимирующей прямой для экстраполяции?
8. Как определяется номер участка с точкой, для которой необходимо выполнить интерполяцию с $h = \text{const}$?
9. Назовите функцию для инициализации графики в языке Паскаль.
10. Как оценивается точность интерполяции в программе *Control*?

ЛЕКЦИЯ 2

КУСОЧНО-ЛИНЕЙНАЯ ИНТЕРПОЛЯЦИЯ С ПЕРЕМЕННЫМ ШАГОМ

Методы аппроксимации кривых намагничивания.

Алгоритм интерполяции и экстраполяции для переменного шага.

Создание функции для интерполяции и экстраполяции на языке Паскаль.

Создание программы для определения точности интерполирования с графической иллюстрацией при разной разбивке интервала на участки.

Содержание лабораторной работы.

2.1. Методы аппроксимации кривых намагничивания

В электромеханике различают **кривые намагничивания электротехнических сталей** и **кривые намагничивания магнитных систем**.

♦ **Кривые намагничивания электротехнических сталей** – это зависимости индукции магнитного поля от напряженности магнитного поля $B = f(H)$.

Кривые намагничивания разных марок стали существенно отличаются друг от друга. Они зависят от процентного содержания добавок, прежде всего кремния, и от условий изготовления стали. На рис. 2.1 и 2.2 приводятся кривые намагничивания холоднокатаной стали марки 2211 и горячекатанной стали марки 3412.

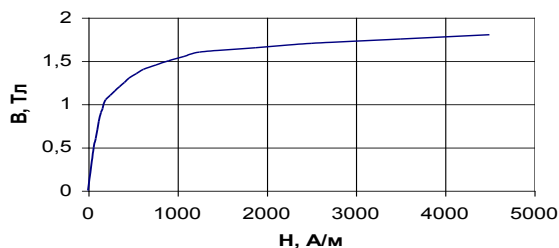


Рисунок 2.1 – Зависимость $B = f(H)$
для стали 3412

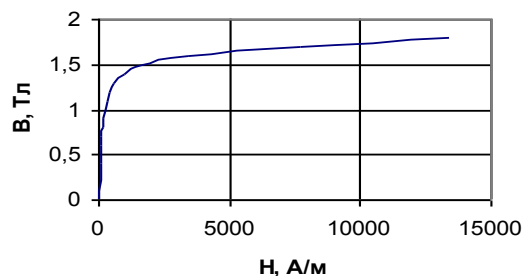


Рисунок 2.2 – Зависимость $B = f(H)$
для стали 2211

На кривых намагничивания различают участки:

- 1) *начальный линейный участок*;
- 2) *участок насыщения*;
- 3) *колесо кривой* – переход от линейного участка к участку насыщения.

Вопросу аппроксимации кривых намагничивания электротехнических стале́й посвящено большое количество работ. В большинстве работ используются зависимости в виде полиномов различных порядков, а также гиперболический синус

$$H = \alpha \cdot sh(\beta B).$$

Однако аналитические зависимости могут быть использованы только для качественного анализа процессов в машине, так как не обеспечивают достаточную точность расчетов.

Поэтому для расчетов современных высокоиспользуемых электрических машин с насыщенными магнитными системами для аппроксимации кривых намагничивания электротехнических стале́й широко применяется кусочная интерполяция: кусочно-линейная и интерполяция сплайнами.

♦ ***Кривая намагничивания магнитной системы*** – это зависимость потока или потокосцепления от тока, например $\Psi = f(I)$.

Она зависит от размеров магнитной системы, величины воздушных зазоров и кривых намагничивания электротехнических стале́й, из которых изготовлена магнитная система. Кривые $\Psi = f(I)$ отличаются от кривых $B = f(H)$ большими размерами линейного начального участка.

Для примера на рис. 2.3 приведены кривые намагничивания вентиляно-индукторного двигателя при разных взаимных положениях полюсов ротора и статора (от согласованного положения при максимальном потокосцеплении до рассогласованного положения при минимальном потокосцеплении).

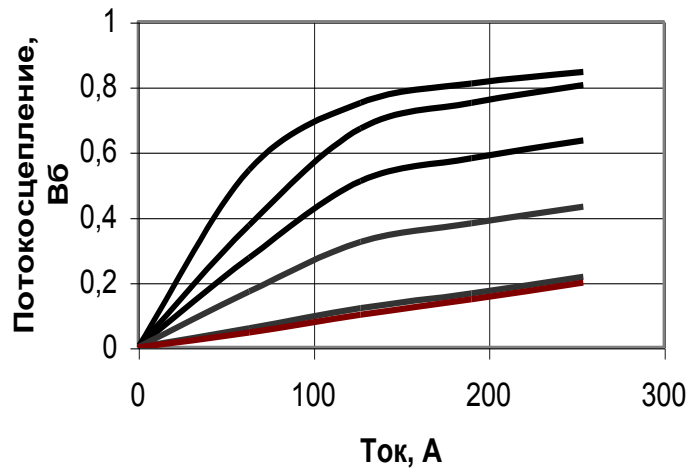


Рисунок 2.3 – Кривые намагничивания ВИД

Для аппроксимации зависимостей $\Psi = f(I, \Theta)$, где Θ – угол между полюсами статора и ротора, иногда используется аналитическое выражение. Однако более высокую точность аппроксимации можно получить путем использования кусочной интерполяции (кусочно-линейной и сплайнов).

РЕЗЮМЕ

Наличие линейного участка в кривых намагничивания делает целесообразным применение кусочно-линейной интерполяции с переменным шагом.

2.2. Алгоритм интерполяции и экстраполяции для переменного шага

Размеры исходной таблицы данных можно уменьшить, если располагать информацией о характере изменения функции в заданном диапазоне. В этом случае шаги между узлами h_i выбирают исходя из критерия

$$\varepsilon_i \approx \varepsilon_{\text{доп}},$$

где $\varepsilon_{\text{доп}}$ – допустимая погрешность аппроксимации, т.е. погрешность на различных участках должна быть примерно одинаковой. Визуально это означает, что прямые линии аппроксимирующей функции примерно одинаково приближаются к исходной кривой на различных участках.

Исходные данные для интерполяции при $h = var$

1. Значения аргументов x_i в узлах таблицы;
2. Значения функции f_i в узлах таблицы;
3. Значение n – число узлов минус 1;
4. Значение $x = t$, для которого необходимо найти приближенное значение функции.

Алгоритм интерполяции при $h = var$

1. Находим номер участка i , на котором находится значение аргумента $x=t$, для которого необходимо найти приближенное значение функции $\varphi(t)$. Для этого в цикле по i сравниваем t и правое значение аргумента данного участка x_i . Если $t > x_i$, переходим к следующему участку. Если $t < x_i$, переходим ко второму пункту;

2. Вычисляем приближенное значение $\varphi(t)$ по формуле интерполяции для переменного шага

$$\varphi(t) = f_{i-1} + (t - x_{i-1}) (f_i - f_{i-1}) / (x_i - x_{i-1}). \quad (2.1)$$

В этой формуле $t - x_{i-1} = \Delta x$, $(f_i - f_{i-1}) / (x_i - x_{i-1}) = tg\alpha_i$ характеризует наклон i -го участка.

Алгоритм экстраполяции при $h = var$

Если $t < x_0$ $\Delta x = t - x_0$; $tg\alpha = (f_1 - f_0) / (x_1 - x_0)$; $\varphi(t) = f_0 + \Delta x \cdot tg\alpha$.

Окончательно $\varphi(t) = f_0 + (t - x_0) \cdot (f_1 - f_0) / (x_1 - x_0)$ (2.2)

Если $t > x_n$ $\Delta x = t - x_n$; $tg\alpha = (f_n - f_{n-1}) / (x_n - x_{n-1})$; $\varphi(t) = f_n + \Delta x \cdot tg\alpha$.

Окончательно $\varphi(t) = f_n + (t - x_n) \cdot (f_n - f_{n-1}) / (x_n - x_{n-1})$. (2.3)

2.3. Создание функции для интерполяции и экстраполяции на языке Паскаль

Если значение аргумента $x = t$ не выходит за пределы таблицы, для определения приближенного значения функции достаточно иметь функцию интерполяции. В соответствии с алгоритмом интерполяции для определения номера участка, на котором находится значение аргумента t , используем цикл с послеусловием repeat-until.

Функция интерполяции для $h = var$

Function FI(t:Real):Real;

Var

i:Integer;

Begin

i:=0;

Repeat i:= i+1 **Until** t<x[i];

FI:=f[I - 1])+(t-x[i-1])*(f[i] - f[i-1])/(x[i] - x[I - 1]);

End;

Если значение аргумента $x=t$ может выходить за пределы таблицы, необходимо использовать функцию интерполяции-экстраполяции.

Функция интерполяции-экстраполяции при $h=var$

Function FI(t:Real):Real;

Var

i:Integer;

Begin

If t<x[0] **Then**

fi:=f[0]+(t-x[0])*(f[1]-f[0])/(x[1]-x[0]) {формула экстраполяции
(2.2)}

Else

If t>x[n] **Then**

fi:=f[n]+(t-x[n])*(f[n]-f[n-1])/(x[n]-x[n-1]) {формула экстраполяции
(2.3)}

Else

Begin

i:=0; **Repeat** i:=i+1 **Until** t<x[i];

fi:=f[i-1]+(t-x[i-1])*(f[i]-f[i-1])/(x[i]-x[i-1]); {формула интерполяции (2.1)}

End;

End;

2.4. Создание программы для определения точности интерполирования с графической иллюстрацией для $h = var$

При создании программы для $h = var$ *CONTROL1* берем за основу программу *CONTROL* при $h = const$, рассмотренной в лекции 1, и вносим в эту программу *следующие изменения*:

1. В описательной части программы в разделе *Const* убираем описание h , x_0 , добавляем массив x и вносим изменения в массив f ;
2. В процедуре *ERROR* заменяем элементы массивов xt , ft и функцию интерполяции-экстраполяции *FI*;
3. В процедуре *VISIOCON* в описательной части убираем описание массива x и в исполнительной части убираем расчет элементов массива x . Полученная таким способом программа содержит таблицы для кривых намагничивания стали 3412. Для других марок стали необходимо изменить элементы массивов x , xt , f , ft . Содержание этих массивов необходимо также изменять при выборе других вариантов разбивки на участки.

2.5. Содержание лабораторной работы

*Ничто в жизни не достается
без большого труда
Гораций (65-8 гг. до н.э.)*

Цели лабораторной работы: 1) ознакомиться с характером кривых намагничивания современных электротехнических сталей, а также с характером кривых намагничивания магнитных систем современных электрических машин; 2) приобрести умение разбивать интервал изменения функции на минимальное число участков при условии обеспечения заданной точности интерполяции; 3) совершенствовать навыки программирования на языке Паскаль.

Для выполнения 1-й и 2-й целей необходимо с помощью программы *CONTROL1* произвести расчеты для нескольких вариантов разбивки на участки для стали 3412, а также для других марок стали, данные для которых предоставит преподаватель. Результаты расчета занести в таблицы. Пример таблицы:

Таблица – Марка стали: сталь....

№ варианта	Значения x_i в узлах таблицы	Число узлов n	ε_{\max}
1			
2			
3			

Проанализировать результаты и выбрать оптимальные варианты для каждой марки стали, обеспечивающие выполнение условия

$$\varepsilon_{\max} < \varepsilon_{\text{доп}}$$

при минимальном числе разбиений на участки.

Аналогичные расчеты необходимо провести для зависимостей $\Psi = f(I)$ и $\Psi = f(\Theta)$ для вентильно-индукторного двигателя по данным, выданным преподавателем.

Для достижения 3-й цели рекомендуется составить программу для автоматического выбора значений элементов массивов x , xt , f , ft с помощью данных, занесенных во внешний файл.

Контрольные вопросы

1. Что такое „кривая намагничивания стали”?
2. Что такое „кривая намагничивания магнитной системы”?
3. Назовите участки кривой намагничивания.
4. Перечислите исходные данные, необходимые для интерполяции с переменным шагом.
5. Как определяется номер участка с точкой, для которой необходимо выполнить интерполяцию с $h = var$?
6. В чем преимущество интерполяции с переменным шагом по сравнению с интерполяцией с постоянным шагом?
7. Какой критерий используется для выбора узлов интерполяции при использовании интерполяции с переменным шагом?

ЛЕКЦИЯ 3

ИНТЕРПОЛЯЦИЯ СПЛАЙНАМИ

Вывод расчетных выражений для коэффициентов сплайнов.

Алгоритм интерполяции сплайнами и создание процедуры и функции для его реализации.

Создание программы для оценки точности интерполирования с графической иллюстрацией.

Примеры применения сплайнов в задачах электромеханики.

3.1. Вывод расчетных выражений для коэффициентов сплайнов

Сплайн – кубический полином, с помощью которого функция аппроксимируется на интервале $[x_i - x_{i-1}]$:

$$\varphi_i(x) = a_i + b_i \cdot (x - x_{i-1}) + c_i \cdot (x - x_{i-1})^2 + d_i \cdot (x - x_{i-1})^3; \quad i=1, \dots, n. \quad (3.1)$$

Коэффициенты a_i, b_i, c_i, d_i определяют с помощью **условий сшивания**.

Два условия получим исходя из критерия: значение аппроксимирующей функции и реальной в узлах слева и справа должны совпадать.

$$1. \quad x = x_{i-1}; \quad \varphi_i(x_{i-1}) = a_i = f_{i-1}; \quad a_i = f_{i-1}; \quad (3.2)$$

$$2. \quad x = x_i; \quad x_i - x_{i-1} = h_i; \quad \varphi_i(x_i) = a_i + b_i \cdot h_i + c_i \cdot h_i^2 + d_i \cdot h_i^3 = f_i; \quad (3.3)$$

Следующие два уравнения получаем из условия непрерывности 1-й и 2-й производных в узлах:

$$\varphi_i'(x_i) = \varphi_{i+1}'(x_i); \quad \varphi_i''(x_i) = \varphi_{i+1}''(x_i).$$

Из уравнения (3.1) определим производные

$$\varphi_i'(x) = b_i + 2 \cdot c_i \cdot (x - x_{i-1}) + 3 \cdot d_i \cdot (x - x_{i-1})^2; \quad \varphi_i''(x) = 2 \cdot c_i + 6 \cdot d_i \cdot (x - x_{i-1});$$

$$\varphi_i'(x_i) = b_i + 2 \cdot c_i \cdot h_i + 3 \cdot d_i \cdot h_i^2; \quad \varphi_i''(x_i) = 2 \cdot c_i + 6 \cdot d_i \cdot h_i;$$

$$\varphi_{i+1}'(x_i) = b_{i+1}; \quad \varphi_{i+1}''(x_i) = 2 \cdot c_{i+1}.$$

Окончательно получаем еще два уравнения:

$$b_i + 2 \cdot c_i \cdot h_i + 3 \cdot d_i \cdot h_i^2 = b_{i+1}; \quad (3.4)$$

$$c_i + 3 \cdot d_i \cdot h_i = c_{i+1}. \quad (3.5)$$

Учитываем условия на концах: принимаем, что за пределами интервала функция изменяется по линейному закону. Тогда

$$\varphi_I''(x_0) = 0; \quad 2 \cdot c_1 + 6 \cdot d_1 \cdot (x_0 - x_0) = 0; \quad c_I = 0; \quad (3.6)$$

$$\varphi_n''(x_n) = 0; \quad c_n + 3 \cdot d_n \cdot h_n = 0. \quad (3.7)$$

Для решения уравнений (3.2) – (3.7) выражаем коэффициенты a_i , b_i , d_i через c_i . Из (3.5) получим

$$d_i = (c_{i+1} - c_i) / (3 \cdot h_i). \quad (3.8)$$

Вычитаем из уравнения (3.3) уравнение (3.2), подставляем (3.8) и получаем выражение для коэффициента b_i :

$$b_i = (f_i - f_{i-1}) / h_i - (c_{i+1} + 2 \cdot c_i) \cdot h_i / 3. \quad (3.9)$$

Подставляем (3.8) и (3.9) в (3.4) и после преобразований получаем уравнение

$$c_{i+2} \cdot h_{i+1} + 2 \cdot (h_{i+1} + h_i) \cdot c_{i+1} + c_i \cdot h_i = 3 \cdot ((f_{i+1} - f_i) / h_{i+1} - (f_i - f_{i-1}) / h_i);$$

$$i=2, \dots, n.$$

Уменьшаем все индексы на 1

$$c_{i+1} \cdot h_i + 2 \cdot (h_i + h_{i-1}) \cdot c_i + c_{i-1} \cdot h_{i-1} = 3 \cdot ((f_i - f_{i-1}) / h_i - (f_{i-1} - f_{i-2}) / h_{i-1});$$

$$i=2, \dots, n. \quad (3.10)$$

Система уравнений (3.10) имеет трехдиагональную матрицу коэффициентов. Эффективным методом для решения таких уравнений является **метод прогонки**.

Метод прогонки состоит из двух этапов:

Первый этап: определяют рекуррентные соотношения между значениями коэффициентов c соседних сплайнов $c_i = f(c_{i+1})$;

Второй этап, так называемый обратный ход: используя рекуррентные соотношения находим коэффициенты c_i .

Для вывода рекуррентных соотношений представим систему уравнений (3.10) в следующем виде:

$$c_{i+1} \cdot h_i + s_i \cdot c_i + c_{i-1} \cdot h_{i-1} = r_i; \quad i=2, \dots, n; \quad (3.11)$$

$$s_i = 2 \cdot (h_i + h_{i-1}); \quad (3.12)$$

$$r_i = 3 \cdot ((f_{i+1} - f_i) / h_{i+1} - (f_i - f_{i-1}) / h_i). \quad (3.13)$$

Запишем (3.11) для $i=2$:

$$c_3 \cdot h_2 + s_2 \cdot c_2 + c_1 \cdot h_1 = r_2; \quad c_1 = 0; \quad c_2 = r_2 / s_2 - h_2 / s_2 \cdot c_3; \quad c_2 = k_2 - l_2 \cdot c_3.$$

Полагаем $i = 3$:

$$c_4 \cdot h_3 + s_3 \cdot c_3 + h_2 \cdot c_2 = r_3; \quad c_4 \cdot h_3 + s_3 \cdot c_3 + h_2 \cdot k_2 - h_2 \cdot l_2 \cdot c_3 = r_3.$$

$$\text{Отсюда } c_3 = (r_3 - h_2 \cdot k_2) / (s_3 - h_2 \cdot l_2) - h_3 / (s_3 - h_2 \cdot l_2) \cdot c_4; \quad c_3 = k_3 - l_3 \cdot c_4.$$

Окончательно на основании последних выражений запишем формулы для **коэффициентов прогонки**

$$k_i = (r_i - h_{i-1} \cdot k_{i-1}) / (s_i - h_{i-1} \cdot l_{i-1}); \quad l_i = h_i / (s_i - h_{i-1} \cdot l_{i-1}); \quad (3.14)$$

$$\text{и рекуррентное соотношение} \quad c_i = k_i - l_i \cdot c_{i+1}. \quad (3.15)$$

Из соотношения (3.15) при $i = 1$ $c_1 = k_1 - l_1 \cdot c_2 = 0$ получим

$$k_1 = 0; \quad l_1 = 0. \quad (3.16)$$

Полагаем $i = n$; $c_n = k_n - l_n \cdot c_{n+1}$; $c_{n+1} = 0$, так как за пределами интервала функция изменяется по линейному закону. Получаем соотношение

$$c_n = k_n. \quad (3.17)$$

РЕЗЮМЕ

Умение выводить расчетные соотношения для коэффициентов сплайнов позволит избежать ошибок при составлении программ.

3.2. Алгоритм интерполяции сплайнами и создание процедуры и функции для его реализации.

На основании полученных расчетных соотношений для коэффициентов сплайнов составляем следующий **алгоритм интерполяции сплайнами**:

1. Предварительно рассчитываем коэффициенты c_i для всех сплайнов по формулам (3.12) – (3.17);
2. Определяем номер участка, на котором находится значение аргумента $x = t$, для которого определяется приближенное значение функции, т.е. номер сплайна i ;
3. Рассчитываем коэффициенты a_i , b_i , d_i по формулам (3.2, 3.8, 3.9) и затем по соотношению (3.1) находим приближенное значение функции.

Первый пункт алгоритма реализуем при составлении программы в процедуре с названием *PROGON*. Для 2-го и 3-го пунктов составляем функ-

цию *FISP*. Далее приводим описание процедуры и функции с комментариями.

Procedure PROGON(**Var** h,c:Array Of **Real**);

Var

i,j,m:**Integer**;

l,k:Array[1..10] Of **Real**; {описание массивов коэффициентов прогонки}

r,s:**Real**;

Begin

k[1]:=0; l[1]:=0; c[1]:=0;

For i=1 **To** n **Do** {расчет массивов коэффициентов прогонки}

Begin

j:=i-1; m:=j-1;

h[i]:=x[i]-x[j]; h[j]:=x[j]-x[m]; {расчет шагов для соседних сплайнов}

s:=2*(h[j]+h[i]); r:=3*((f[i]-f[j])/h[i]-(f[j]-f[m])/h[j]);

k[i]:=(r-h[j]*k[j])/(s-h[j]*l[j]); l[i]:=h[i]/(s-h[j]*l[j]);

End;

{расчет массива коэффициентов сплайнов при квадрате аргумента}

c[n]:=k[n];

For i:=n-1 **Downto** 2 **Do** c[i]:=k[i]-l[i]*c[i+1]; {рекуррентное соотношение}

End;

Function FISP(t:real): real;

Var

i:**Integer**;

a,b,d,p:**Real**;

Begin

i:=0; **Repeat** i:=i+1 **Until** (t<x[i]) or (i=n); {определение номера сплайна} {расчет коэффициентов сплайнов}

```

a:=f[i-1]; d:=(c[i+1]-c[i])/(3*h[i]); b:=(f[i]-f[i-1])/h[i]-
(c[i+1]+2*c[i])*h[i]/3;      {определение приближенного значения функ-
ции для аргумента x=t по выражению (3.1)}
p:=t-x[i-1]; FISP:=a+b*p+c[i]*p*p+d*p*p*p;
End;

```

3.3. Создание программы для оценки точности интерполирования с графической иллюстрацией

Составляем программу, которая позволяет сравнить метод кусочно-линейной интерполяции с переменным шагом и метод интерполяции сплайнами. Поэтому берем за основу программу *CONTROL1* для кусочно-линейной интерполяции с $h = var$ и вносим следующие изменения:

1. В головной модуль добавляем описание массивов h и c , описание процедуры *PROGON* и функции *FISP*;

2. В процедуру *ERROR* добавляем расчет погрешности для сплайнов

```
esp:=ABS((ft[i]-FISP(xt[i]))/ft[i]),
```

описание переменной esp и вывод на печать погрешностей для двух методов в сравнении для каждого значения $xt[i]$:

```
Writeln('xt[i]=' , xt[i]);
```

```
Writeln('кус.-лин.', ' e=', e:5:3, ' сплайн', ' esp=', esp:5:3);
```

3. В процедуру *VISIOCON* добавляем процедуру *COORDSPL*, которая позволяет рассчитать значения координат сплайнов для построения аппроксимирующей кривой. Каждый интервал, соответствующий сплайну, делим на ks отрезков.

```
Procedure COORDSPL(Var xsp,fsp:Array Of Real);
```

```
  Const ks=5;
```

```
  Var
```

```
    i, m: Integer;
```

```
  Begin
```

```
    l:=-1;
```

```
    For i:=1 To n Do
```

```
      For m:=1 To ks Do
```

Begin

$l:=l+1;$

$xsp[l]:=x[i-1]+(m-1)*h[i]/ks; fsp[l]:=fisp(xsp[l]);$

End;

$xsp[l+1]:=x[n]; fsp[l+1]:=f[n];$

End;

Переменную l и массивы xsp , fsp описываем в описательной части процедуры *VISIOCON*. В процедуре *GRAPHIKA* добавляем операторы для построения графика по координатам, рассчитанным в процедуре *COORDSPL*.

Обращение к процедуре *COORDSPL* помещаем перед обращением к процедуре *GRAPHIKA*.

В головном модуле вызываем процедуры в следующем порядке:

PROGON; ERROR; VISIOCON.

3.4. Примеры применения сплайнов в задачах электромеханики

В последние годы метод интерполяции сплайнами получил широкое распространение, во многих задачах он заменил метод кусочно-линейной интерполяции.

Пример 1: Задача расчета магнитного поля электрических машин численными методами. Раньше для расчета магнитного поля применяли метод конечных разностей (сеток), в котором для интерполяции кривых намагничивания электротехнических сталей использовали кусочно-линейную интерполяцию. В настоящее время для расчета магнитного поля электрических машин используют метод конечных элементов. Широкое распространение получили пакеты прикладных программ, такие как **ANSYS** – для расчета трехмерных и двумерных полей, **FEM** – для расчета двумерных полей, **ELCUT** – для расчета двумерных полей. Во всех этих программах для интерполяции кривых намагничивания электротехнических сталей применяют сплайны.

Пример 2: Применение сплайнов для интерполяции кривых намагничивания магнитной системы вентильно-индукторных двигателей при раз-

ном положении полюсов статора и ротора, а также для интерполяции зависимости потокосцепления от угла между полюсами.

3.5. Содержание лабораторной работы

*Нет ничего ужаснее деятельного
невежества.*

Гете 1749-1832 г.

В соответствии с приведенной цитатой рекомендуется перед выполнением лабораторной работы проверить свои знания по теме, а затем выполнить следующее:

1. Скомпоновать программу для сравнения методов интерполяции: сплайнами и кусочно-линейной интерполяции с переменным шагом в соответствии с указаниями подраздела 3.3;

2. Провести расчеты по этой программе для разных марок стали при разной разбивке на интервалы и результаты занести в таблицы, аналогичные табл.3.1.

Таблица 3.1 – Результаты расчета для стали.....

№ варианта	Значения x_i в узлах таблицы	Число узлов n	ε_{\max} сплайны	ε_{\max} кус.лин.
1				
2				
3				

По данным таблицы делаем вывод об эффективности методов интерполяции кусочно-линейной с переменным шагом и сплайнами.

3. Добавить в программу операторы для определения ε_{\max} для сплайнов и для кусочно-линейной интерполяции.

Контрольные вопросы

1. Дайте определение понятию „сплайны”.
2. Запишите математическое выражение для сплайна.

3. Перечислите условия „сшивания” сплайнов, которые используются для определения коэффициентов сплайнов.
4. Какой коэффициент сплайна определяется путем решения системы уравнений?
5. Суть метода прогонки для решения системы уравнений.
6. Как называются соотношения между соседними значениями коэффициентов сплайнов при x^2 ?
7. Перечислите последовательность определения коэффициентов сплайнов в программе.
8. Запишите математическое выражение для рекуррентного соотношения.
9. Что такое „коэффициенты прогонки”?
10. Назовите примеры применения сплайнов в электромеханике.

ЛЕКЦИЯ 4

ИНТЕРПОЛИРОВАНИЕ ФУНКЦИЙ ДВУХ ПЕРЕМЕННЫХ С ПОМОЩЬЮ МАТЕМАТИЧЕСКОГО ПЛАНИРОВАНИЯ ЭКСПЕРИМЕНТА (ПОЛНЫЙ ФАКТОРНЫЙ ЭКСПЕРИМЕНТ)

Термины и определения теории планирования эксперимента.

Описание физики задачи, для которой создается модель.

Построение программы.

Содержание лабораторной работы.

4.1. Термины и определения теории планирования эксперимента

Применительно к задаче интерполирования функций *эксперимент* – это расчет или определение из таблиц или кривых (графиков) значений функции в узлах интерполирования для значений аргументов, которые здесь задаются по специальному *плану эксперимента*.

В результате обработки данных эксперимента строится *математическая модель* – аппроксимирующее выражение, с помощью которого и определяются значения функции между узлами интерполяции. Находят применение следующие *математические модели*:

1) *неполная квадратичная модель*

$$\varphi(x_1, x_2) = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_1 \cdot x_2; \quad (4.1)$$

2) *квадратичная модель*

$$\varphi(x_1, x_2) = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_1^2 + b_4 \cdot x_2^2 + b_5 \cdot x_1 \cdot x_2. \quad (4.2)$$

Здесь x_1, x_2 – независимые переменные, называются *факторами*. Прописные буквы X_1, X_2 обозначают реальные факторы, строчные буквы x_1, x_2 – кодированные факторы. Обработка эксперимента проводится обычно для кодированных факторов. Связь между реальными и кодированными факторами устанавливают соотношения

$$x_1 = (X_1 - X_{1cp}) / I_1; \quad x_2 = (X_2 - X_{2cp}) / I_2; \quad (4.3)$$

$$X_1 = x_1 \cdot I_1 + X_{1cp}; \quad X_2 = x_2 \cdot I_2 + X_{2cp}. \quad (4.4)$$

Здесь I_1, I_2 – интервалы изменения факторов, равные половине диапазона изменения факторов

$$I_1 = (X_{1\max} - X_{1\min})/2; \quad I_2 = (X_{2\max} - X_{2\min})/2. \quad (4.5)$$

X_{1cp}, X_{2cp} – средние значения факторов в заданном диапазоне их изменения

$$X_{1cp} = (X_{1\max} + X_{1\min})/2; \quad X_{2cp} = (X_{2\max} + X_{2\min})/2.$$

Кодирование означает перенос начала координат в середину диапазона изменения факторов, в точку с координатами X_{1cp}, X_{2cp} . При этом

$$x_{1\max} = +1; \quad x_{1\min} = -1; \quad x_{2\max} = +1; \quad x_{2\min} = -1; \quad x_{1cp} = 0; \quad x_{2cp} = 0.$$

План эксперимента для неполной квадратичной модели для двух факторов и его **геометрическая интерпретация** имеют следующий вид (табл. 4.1):

Таблица 4.1 – План эксперимента для полного факторного эксперимента

№ опыта	x_1	x_2
1	-1	-1
2	+1	-1
3	-1	+1
4	+1	+1

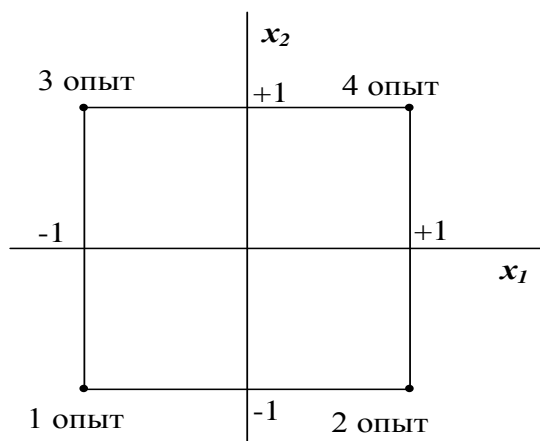


Рисунок 4.1 – Геометрический образ плана ПФЭ 2^2

Эксперимент по этому плану называется **полным факторным экспериментом ПФЭ 2^2** . Здесь степень означает число факторов, основание – число уровней факторов.

Коэффициенты модели определяют по **методу наименьших квадратов** исходя из критерия: сумма квадратов отклонений опытных значений функции Y_u и значений функции, рассчитанных по уравнению модели $\varphi(x_1, x_2)$, должна быть минимальной. На основании этого критерия для ПФЭ 2^2 получено соотношение

$$b_i = \Sigma x_{iu} \cdot Y_u / 4. \quad (4.6)$$

Значения x_{iu} определяют из **матрицы планирования**. Ниже приводятся **матрица планирования** и вектор-столбец опытных значений функции в точках плана эксперимента. Здесь x_0 – фиктивный фактор, тождественно равный 1,

$x_3 = x_1 \cdot x_2$, u - номер опыта.

Таблица 4.2 – Матрица планирования

№ опыта	$i=0$ $x_0=1$	$i=1$ x_1	$i=2$ x_2	$i=3$ $x_3=x_1 \cdot x_2$	Y_u
1	+1	-1	-1	+1	Y_1
2	+1	+1	-1	-1	Y_2
3	+1	-1	+1	-1	Y_3
4	+1	+1	+1	+1	Y_4

Для **ПФЭ** опытные значения функции и рассчитанные по уравнению модели в точках плана эксперимента точно совпадают. Поэтому для оценки точности аппроксимации проводят дополнительный опыт в нулевой точке для кодированных факторов $x_1 = 0, x_2 = 0$ (в средней точке для реальных факторов), из опыта определяют Y_0 . Из уравнения модели (4.1) получаем $\varphi(0,0) = b_0$. Погрешность аппроксимации определяем по соотношению

$$\varepsilon = |(Y_0 - b_0) / Y_0|. \quad (4.7)$$

Адекватность модели оцениваем по критерию

$$\varepsilon < \varepsilon_{\text{доп}} , \quad (4.8)$$

где $\varepsilon_{\text{доп}}$ – допустимая погрешность аппроксимации.

4.2. Описание физики задачи, для которой создается математическая модель

Важными вопросами при проектировании электротехнических устройств являются вопросы определения коэффициента полезного действия (КПД) и максимального превышения температуры обмоток над температурой окружающей среды. Для определения этих параметров необходимо рассчитать все виды потерь в устройстве, в том числе, магнитные потери – потери в магнитной системе. Эти потери разделяют на потери от гистерезиса и от вихревых токов. На практике при частоте 50 Гц обычно определяют магнитные потери, не разделяя их, и пользуются экспериментально установленной зависимостью между индукцией и удельными потерями в стали для данной марки в единице массы $P_{\text{ст}}$ (Вт/кг). Данные экспериментального исследования стали сводятся в таблицы.

Поставим задачу построить неполную квадратичную модель для зависимости удельных потерь в стали от индукции и марки электротехнической стали- $P_{\text{ст}} = f(B, \text{марка стали})$. В этой задаче 2 фактора:

- 1) фактор X_1 – индукция B , Тл;
- 2) фактор X_2 – марка стали.

Это качественный фактор. Для его определения используется ранговый подход. Выбираем 3 марки стали и присваиваем им числа: 1) сталь 3404 толщиной 0,35 мм ; $X_2 = +1$; 2) сталь 3404 толщиной 0,3 мм; $X_2 = 0$; 3) сталь 3405 толщиной 0,3 мм; $X_2 = -1$. Интервал изменения фактора $I_2 = 1$. Эти марки стали применяют для магнитных систем силовых трансформаторов. Необходимо подобрать диапазон изменения 1 фактора - индукции, который обеспечит выполнение критерия адекватности в средней точке

$$\varepsilon < \varepsilon_{\text{доп}} .$$

Рекомендуемая при проектировании индукция в магнитной системе трансформатора в номинальном режиме находится в пределах (1,55 – 1,65) Тл. Поэтому выбираем среднее значение индукции 1,6 Тл, т.е. $X_{1\text{ср}} = 1,6$, а

минимальный интервал изменения фактора $I_1 = 0,05$. Однако с учетом возможного отклонения индукции от рекомендуемых значений желательно определить максимально возможный интервал, при котором выполняется критерий адекватности. Для построения модели используем данные, приведенные в табл. 4.3.

Таблица 4.3 - Удельные потери в стали

B , Тл	сталь 3404, толщина 0,35 мм	сталь 3404, толщина 0,3 мм	сталь 3405, толщина 0,3 мм
1,48	1,066	0,999	0,943
1,5	1,1	1,03	0,97
1,52	1,134	1,07	1,004
1,54	1,168	1,11	1,038
1,56	1,207	1,15	1,074
1,58	1,251	1,19	1,112
1,6	1,295	1,23	1,15
1,62	1,353	1,278	1,194
1,64	1,411	1,326	1,238
1,66	1,472	1,38	1,288
1,68	1,536	1,44	1,344
1,7	1,6	1,5	1,4
1,72	1,672	1,56	1,46

РЕЗЮМЕ

Применение неполной квадратичной модели по уравнению (4.1) вместо использования таблиц существенно упрощает процесс проектирования.

4.3. Построение программы

Составим программу, которая позволяет построить адекватную неполную квадратичную модель для расчета удельных потерь в стали в функции от индукции и марки стали. Программа включает следующие структурные блоки:

1. Функцию *FUNCTION F*(X_1, X_2) для определения удельных потерь по таблице;
2. Процедуру *PROCEDURE EKSPERIMENT* для определения опытных значений функции в точках плана эксперимента;
3. Процедуру *PROCEDURE KOEF* для расчета коэффициентов по результатам эксперимента;
4. Процедуру *PROCEDURE INTERVAL* для определения максимальной величины интервала изменения 1-го фактора, при котором выполняется условие адекватности.

Вводим массивы одномерные для коэффициентов модели $b[0..3]$ и результатов эксперимента $y[0..3]$ и двумерный для матрицы планирования $x[0..3,0..3]$. Все структуры и идентификаторы описываем в головном модуле. Ниже приводится текст программы с комментариями.

Program PFE;

Uses Crt;

Type sqw=**Array**[0..3,0..3] **of Real**;

Const x=sqw=((1,-1,-1,1),(1,1,-1,-1),(1,-1,1,-1),(1,1,1,1));

Var

X1,X2,x1cp,x2cp,I1,I2,pst,y0,edop,erac,b0:Real;

b,y:array[0..3] **of Real**;

Stal:string[10];

i,j:integer;

Function F(**X1,X2:Real**):**Real**;

Begin

If X2= -1 **Then** Stal:='ct3405/.30';

If X2=1 **Then** Stal:='ct3404/.35';

Writeln('индукция B=',X1:5:2,' марка стали=',Stal);

```

    Writeln ('введите потери в стали из таблицы');
    Readln(pst); F:=pst;
End;
Procedure EKSPERIMENT(Var y:Array of Real);
Begin
    For i:=0 to 3 Do Begin
        X1:=x[i,1]*I1+x1cp; X2:=x[i,2]*I2+x2cp;
        y[i]:=F(X1,X2);End;
    End;
Procedure KOEF(Var b:Array of Real);
Begin
    For i:=0 to 3 do Begin
        b[i]:=0;
        For j:=0 to 3 do b[i]:=b[i]+x[j,i]*y[j];{уравнение(4.6)}
        b[i]:=b[i]/4;End;
    End;
Procedure INTERVAL;{подбор максимального интервала}
Begin
    x2cp:=0; I2:=1;{среднее значение 2 фактора- сталь 3404/0.3}
    Writeln('x1cp,Edop,y0 в средней точке');
    Readln(x1cp,Edop,y0);{ввод B=1.6,Edop=0.01,Pст=1.23}
    Erac:=100;
    While Erac>Edop do{цикл с предусловием}
    Begin
        Writeln('введите интервал для 1 фактора I1');
        Readln(I1);{ввод интервала 1 фактора, вначале 0.12}
        EKSPERIMENT(y);{обращение к процедуре}
        b0:=(y[0]+y[1]+y[2]+y[3])/4;
        Erac:=Abs((y0-b0)/b0);{расчетная погрешность}
        Writeln('Edop=',Edop:7:5,' Erac=',Erac:7:5);
        Writeln('если Erac>Edop уменьшайте интервал');
    End;

```

```

End;
Begin{головной модуль- исполнительная часть}
  ClrScr;
  INTERVAL;
  KOEF(b);{расчет коэффициентов модели}
  Writeln('Коэффициенты модели':20);
  For i:=0 to 3 do Writeln('I=',I:2,'  b[I]', b[I]:10:3);
End.

```

4.4. Содержание лабораторной работы

Знание-орудие, а не цель.

Л.Н.Толстой

1828-1910 г.

С целью усвоения теоретического материала лекции, приобретения навыков построения адекватных моделей и с целью совершенствования навыков программирования на языке Паскаль выполните следующие задания:

1.С помощью приведенной в подразделе 4.3. программы и данных таблицы определите коэффициенты адекватной неполной квадратичной модели зависимости удельных потерь в стали от индукции и марки стали для максимально допустимого интервала изменения 1- го фактора;

2. Напишите процедуру для определения погрешности интерполяции в промежуточных точках при изменении индукции от B_{\max} до B_{\min} для каждой марки стали;

3. Напишите процедуру для визуального контроля адекватности модели с помощью графического построения двух зависимостей $P_{\text{ст}} = f(B)$ для каждой марки стали, построенных по уравнению модели и по данным таблицы.

Контрольные вопросы

1. Какие типы моделей применяются в математическом планировании эксперимента?

2. Что такое „факторы”?
3. Запишите уравнение связи между кодированными и реальными факторами.
4. Суть кодирования факторов.
5. План эксперимента для неполной квадратичной модели для двух факторов.
6. Геометрическая интерпретация плана эксперимента для неполной квадратичной модели.
7. Что такое „полный факторный эксперимент”?
8. Матрица планирования для полного факторного эксперимента ПФЭ².
9. Формула для расчета коэффициентов модели и критерий для их определения.
10. Как определяется адекватность модели для полного факторного эксперимента?
11. Опишите физику задачи, для которой необходимо было построить математическую модель на лабораторном занятии.

ЛЕКЦИЯ 5
ИНТЕРПОЛИРОВАНИЕ ФУНКЦИЙ ДВУХ ПЕРЕМЕННЫХ
С ПОМОЩЬЮ МАТЕМАТИЧЕСКОГО ПЛАНИРОВАНИЯ
ЭКСПЕРИМЕНТА
(ОРТОГОНАЛЬНЫЙ ЦЕНТРАЛЬНО-КОМПОЗИЦИОННЫЙ ПЛАН
ВТОРОГО ПОРЯДКА)

План эксперимента, матрица планирования, уравнение модели и формула для расчета коэффициентов для ортогональных центрально-композиционных планов 2-го порядка.

Программа для построения модели.

Примеры применения математического планирования эксперимента в задачах электромеханики.

Содержание лабораторной работы.

5.1. План эксперимента, матрица планирования, уравнение модели и формула для расчета коэффициентов для ортогональных центрально-композиционных планов 2-го порядка

Ортогональный центрально-композиционный план (ОЦКП) 2-го порядка позволяет получить квадратичную модель в виде

$$\varphi(x_1, x_2) = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_1^2 + b_4 \cdot x_2^2 + b_5 \cdot x_1 \cdot x_2. \quad (5.1)$$

План эксперимента ***ОЦКП 2-го порядка*** включает:

1. 4 точки по плану полного факторного эксперимента;
2. 4 "звездные" точки $x_i = +1$ при $x_j = 0$ и $x_i = -1$ при $x_j = 0$, $i, j = 1, 2$;
3. Нулевая точка $x_i = 0$; $i = 1, 2$.

Ниже приведены план эксперимента для кодированных факторов и его геометрический образ (табл. 5.1, рис. 5.1).

Таблица 5.1 – План эксперимента для ОЦКП 2- го порядка

№ опыта	x_1	x_2
1	-1	-1
2	1	-1
3	-1	1
4	1	1
5	-1	0
6	1	0
7	0	-1
8	0	1
9	0	0

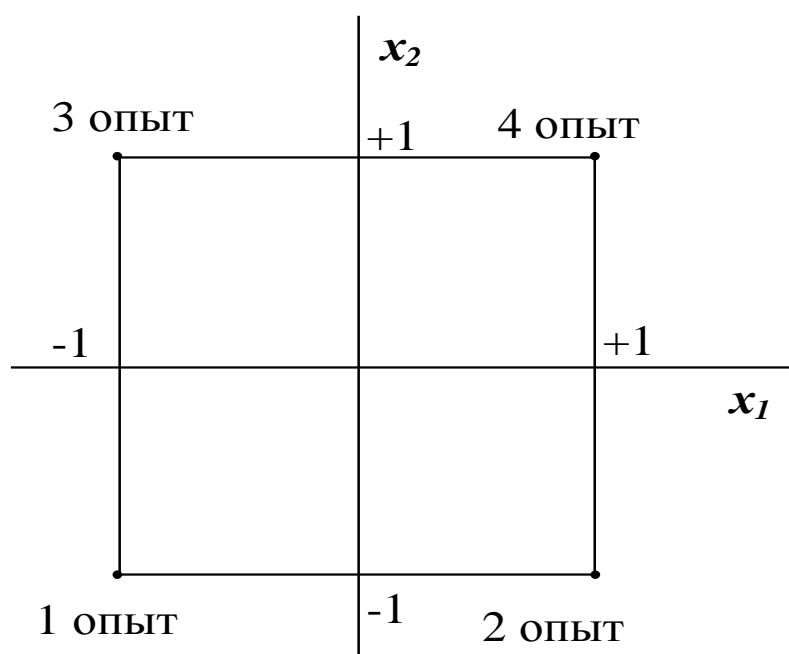


Рисунок 5.1 - Геометрический образ плана ОЦКП 2-го порядка

Рассмотрим матрицу планирования, которая включает кроме плана эксперимента столбцы, учитывающие нулевой фактор, квадраты факторов и взаимодействия факторов, т.е. их произведение. Квадраты факторов преобразуются, так как из условия ортогональности плана требуется, чтобы сумма элементов любого столбца была равна нулю.

$$x'_i = x^2_i - x^2_{icp}; \quad x^2_{icp} = \sum x^2_i / 9; \quad x^2_{1cp} = x^2_{2cp} = 2/3;$$

Таблица 5.2 – Матрица планирования ОЦКП 2-го порядка
для двух факторов

№ опыта	x_0	x_1	x_2	x'_1	x'_2	$x_1 \cdot x_2$
1	1	-1	-1	1/3	1/3	1
2	1	1	-1	1/3	1/3	-1
3	1	-1	1	1/3	1/3	-1
4	1	1	1	1/3	1/3	1
5	1	-1	0	1/3	-2/3	0
6	1	1	0	1/3	-2/3	0
7	1	0	-1	-2/3	1/3	0
8	1	0	1	-2/3	1/3	0
9	1	0	0	-2/3	-2/3	0

По результатам опытов, проведенных по плану эксперимента, и по данным матрицы планирования эксперимента рассчитывают коэффициенты модели по формуле:

$$b_i = \Sigma x_{iu} \cdot Y_u / \Sigma x_{iu}^2. \quad (5.2)$$

Здесь u – номер опыта.

Уравнение модели, полученное с помощью матрицы планирования, имеет вид

$$\varphi(x_1, x_2) = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot (x_1^2 - 2/3) + b_4 \cdot (x_2^2 - 2/3) + b_5 \cdot x_1 \cdot x_2. \quad (5.3)$$

Преобразуем его к виду уравнения (5.1)

$$\varphi(x_1, x_2) = b'_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_1^2 + b_4 \cdot x_2^2 + b_5 \cdot x_1 \cdot x_2. \quad (5.4)$$

$$\text{Здесь } b'_0 = b_0 - 2/3 \cdot b_3 - 2/3 \cdot b_4. \quad (5.5)$$

РЕЗЮМЕ

Рекомендуется освоить методику построения квадратичных моделей, так как большинство зависимостей в электромеханике имеет существенно нелинейный характер.

5.2. Программа для построения модели

Составим программу для построения адекватной квадратичной модели для зависимости потерь в стали от индукции и от марки стали по данным таблицы, приведенной в лекции 4. При использовании **ОЦКП 2-го порядка** получаем модель, свойства которой отличаются от свойств модели по плану **ПФЭ2²**. Значения функции, рассчитанные по уравнению (5.4) в точках плана эксперимента, не равны опытным значениям функции. Поэтому точность аппроксимации будем оценивать по значению максимальной погрешности

$$\varepsilon_{\max} = \max((\varphi(x_1, x_2) - Y_{\text{оп}}) / Y_{\text{оп}}), \quad (5.6)$$

а адекватность модели будем проверять по критерию

$$\varepsilon_{\max} < \varepsilon_{\text{доп}}. \quad (5.7)$$

Для расчета погрешности создаем процедуру *ERROR*. В остальном структура программы *ОСКР* совпадает со структурой программы *PFE*. Далее приводится текст программы с комментариями.

PROGRAM ОСКР;

Uses Crt;

Type

MATR=Array [0..8,0..5] of Real;

Const

x:MATR=((1,-1,-1,1/3,1/3,1),(1,1,-1,1/3,1/3,-1),(1,-1,1,1/3,1/3,-1),
(1,1,1,1/3,1/3,1),(1,-1,0,1/3,-2/3,0),(1,1,0,1/3,-2/3,0),
(1,0,-1,-2/3,1/3,0),(1,0,1,-2/3,1/3,0),(1,0,0,-2/3,-2/3,0));

Var

X1,X2,x1cp,x2cp,I1,I2,pst,edop,eracm:**Real;**

b:Array[0..5] of **Real;**

y:Array[0..8] of **Real;**

Stal:String[10];

i,j:Integer;

Function F(X1,X2:Real):Real;

Begin

If X2= -1 **Then** Stal:='ct3405/.30';

If X2=1 **Then** Stal:='ct3404/.35';

```

If X2=0 Then Stal:='ct3404/.3';
Writeln('индукция B=',X1:5:2,' марка стали=',Stal);
Writeln ('введите потери в стали из таблицы');
Readln(pst);
F:=pst;
End;

Procedure EKSPERIMENT(Var y:Array of Real);
Begin
  For i:=0 to 8 Do
    Begin
      X1:=x[i,1]*I1+x1cp;
      X2:=x[i,2]*I2+x2cp;
      y[i]:=F(X1,X2);
    End;
  End;

Procedure KOEF(Var b:Array of Real);
Var
  s:Real;
Begin
  For i:=0 to 5 Do
    Begin
      s:=0; b[i]:=0;
      For j:=0 to 8 Do
        Begin
          s:=s+x[j,i]*x[j,i];
          b[i]:=b[i]+x[j,i]*y[j];{уравнение (5.6)}
        End;
      b[i]:=b[i]/s;
    End;
  End;

Procedure ERROR(Var Erasm:Real);
Var

```

```

    y1,E1:Array[0..8] of Real;
Begin
    Erasm:=0;
    For i:=0 to 8 Do
        Begin
            y1[i]:=0;
            For j:=0 to 5 Do y1[i]:=y1[i]+x[i,j]*b[j];
            E1[i]:=ABS((y[i]-y1[i])/y[i]);
            If E1[i]>Erasm Then Erasm:=E1[i];
        End;
    End;
Procedure INTERVAL;{подбор максимального интервала}
Begin
    x2cp:=0; I2:=1;{среднее значение 2 фактора- сталь
3404/0.3}
    Writeln('x1cp,Edop');
    Readln(x1cp,Edop);{ввод B=1.6,Edop=0.01}
    Erac:=100;
    While Erasm>Edop Do{цикл с предусловием}
        Begin
            Writeln('введите интервал для 1-го фактора I1');
            Readln(I1);{ввод интервала 1-го фактора, вначале
0.12}
            EKSPERIMENT(y);{обращение к процедуре}
            KOEF(b);
            ERROR (Erasm);
            Writeln('Edop=',Edop:7:4,' Erasm=',Erasm:7:4);
            Writeln('если Erasm>Edop уменьшайте интервал');
        End;
    End;
Begin{головной модуль – исполнительная часть}
    ClrScr;

```

```
INTERVAL;  
Writeln('Коэффициенты модели':20);  
For i:=0 to 5 Do Writeln('I=',I:2,' b[I]', b[I]:7:3);  
End.
```

5.3. Примеры применения математического планирования эксперимента в задачах электромеханики

Математическое планирование эксперимента позволяет получить адекватные математические модели различных явлений и процессов по результатам проведенных опытов при минимальном их количестве. При этом эксперимент может быть реальным, например исследование теплового состояния электрической машины в различных режимах работы, и модельным. Модель может быть физическая или математическая. Расчет по формулам, с помощью стандартных компьютерных программ, определение значений функции из таблиц или графиков относятся к модельному эксперименту. В электромеханике для получения расчетных математических моделей применяют различные виды эксперимента. Приведем примеры.

1. Бывший главный конструктор Ново-Каховского электромашиностроительного завода Яковлев разработал модель зависимости потерь в стали асинхронного двигателя от конструктивных и технологических факторов. Для разработки модели было изготовлено 54 двигателя с параметрами, изменяющимися по специальному плану Бокса-Бенкина.

2. Аспирант кафедры электрических машин НТУ „ХПИ разработал модель зависимости температуры обмотки якоря от частоты вращения и тока якоря для широкорегулируемого двигателя постоянного тока. Для разработки модели было проведено 9 тепловых режимов, каждый длительностью примерно 4,5 ч, по плану ОЦКП 2-го порядка.

3. Для определения влияния геометрии магнитной системы и размещения обмотки возбуждения на величину потока другой аспирант кафедры электрических машин разработал модели с помощью расчетов магнитного поля методом конечных элементов по стандартной программе. Опыты проводились по плану ОЦКП 2 порядка;

4. Бывший заведующий кафедрой электрических машин Московского энергетического института Копылов И.П. считал, что все многочисленные таблицы и графики в учебниках по проектированию электрических машин должны быть заменены уравнениями, полученными с помощью математического планирования эксперимента.

5.4. Содержание лабораторной работы

*Очень трудно избежать двух вещей:
тупоумия, если замкнуться в своей
специальности, и неосновательности,
если выйти за ее пределы.
Гете 1749–1832 г.*

Для того чтобы избежать неосновательности при использовании математического планирования эксперимента для построения моделей, тщательным образом изучите теоретический материал лекций 4 и 5, просмотрите литературу по теме, а затем выполните следующие задания:

1. Постройте квадратичную модель зависимости потерь в стали от индукции и марки стали с помощью программы, приведенной в подразделе 5.2. Сравните квадратичную модель с неполной квадратичной, полученной на предыдущем занятии;

2. Добавьте в программу операторы для вывода на печать погрешностей во всех точках плана, а также процедуру для графического построения зависимостей $P_{ст} = f(B)$ для всех марок стали по данным таблицы и по уравнению модели.

Контрольные вопросы

1. Нарисуйте план эксперимента для ортогонального центрально-композиционного плана 2-го порядка для двух факторов.

2. Изобразите геометрический образ плана ОЦКП 2-го порядка для двух факторов.

3. Изобразите матрицу планирования ОЦКП 2-го порядка для двух факторов.

4. Напишите формулу для расчета коэффициентов квадратичной модели.
5. Как оценивается адекватность модели, построенной с помощью ОЦКП 2-го порядка?
6. Опишите структуру программы *ОСКР*.
7. Как учитывается качественный фактор (марка стали) при построении модели?
8. Как и почему преобразуются столбцы квадратов факторов в матрице планирования ортогонального центрально-композиционного плана 2-го порядка?
9. Напишите уравнение модели, которое получают с помощью эксперимента, проведенного по плану ОЦКП 2-го порядка.

ЛЕКЦИЯ 6

ЧИСЛЕННОЕ РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Метод простых итераций. Расчетные формулы, алгоритм, программа.

Метод Ньютона. Расчетные формулы, алгоритм, программа.

Создание программы расчета индукции и напряженности в зубцах электрической машины.

Содержание лабораторной работы.

6.1. Метод простых итераций.

Расчетные формулы, алгоритм, программа

Численные методы решения нелинейных уравнений применяются, если отсутствуют аналитические выражения для корней уравнения $F(x) = 0$. Методы поиска корней состоят из двух этапов:

1. На первом этапе приближенно находят интервалы, на которых находятся корни;
2. На втором этапе уточняют значения корней.

Рассмотрим методы поиска корней для второго этапа.

Метод простых итераций (метод последовательных приближений) – метод, в котором осуществляется последовательное уточнение первоначального грубого приближения. Каждый шаг в этом приближении называется **итерацией**. Если в процессе последовательных итераций осуществляется приближение к корню, говорят, что процесс итераций сходится. Если нет, то процесс итераций расходится.

Для получения алгоритма метода простых итераций перепишем исходное уравнение

$$F(x) = 0 \tag{6.1}$$

в следующем виде:

$$x = f(x). \tag{6.2}$$

Задаемся начальным приближением x^0 . Определяем первое приближение x^1 по уравнению (6.2) $x^1 = f(x^0)$, затем второе приближение $x^2 = f(x^1)$, и т. д. Окончательно **алгоритм метода простых итераций** имеет вид

$$x^{n+1} = f(x^n).$$

Условие окончания итерационного процесса

$$abs(x^{n+1} - x^n) < \varepsilon_{\text{доп.}} \quad (6.3)$$

Условие сходимости итерационного процесса

$$f'(x) < 1.$$

Составим по рассмотренному алгоритму процедуру *PRIT*.

Procedure PRIT(Var n:Integer;Var x:Real);

Var

L,Eras:**Real**; {идентификатор для хранения x на предыдущей итерации, расчетная погрешность}

Begin

n:=0; {номер итерации}

Repeat {цикл с послеусловием}

L:=x;

x:=F(x); {по алгоритму}

Eras:=ABS(x-L);

n:=n+1;

Until Eras<Edop;

End;

6.2. Метод Ньютона. Расчетные формулы, алгоритм, программа

Для определения x^{n+1} ($n+1$ приближения) проводим касательную к кривой $F(x)$ в точке со значением аргумента x^n . Пересечение касательной с осью абсцисс определяет x^{n+1} (рис. 6.1).

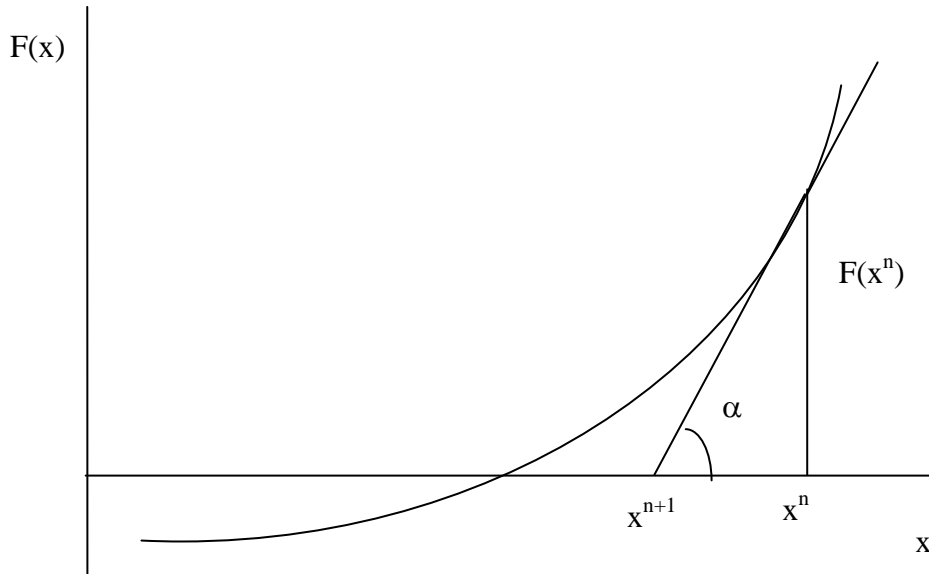


Рисунок 6.1 – Иллюстрация к методу касательных

Из рис. 6.1 следует

$$\operatorname{tg} \alpha = F'(x^n) = F(x^n) / (x^n - x^{n+1}).$$

Отсюда запишем **алгоритм метода Ньютона**

$$x^{n+1} = x^n - F(x^n) / F'(x^n).$$

Если невозможно получить аналитическое выражение для производной, ее вычисляют приближенно по соотношению

$$F'(x^n) = (f(x^n + \varepsilon) - f(x^n)) / \varepsilon;$$

Здесь ε - малая величина. Обычно принимают $\varepsilon = \varepsilon_{\text{доп}}$. Подставляем значение производной в исходную формулу алгоритма и получаем **алгоритм модифицированного метода Ньютона**:

$$x^{n+1} = x^n - F(x^n) \cdot \varepsilon_{\text{доп}} / (F(x^n + \varepsilon_{\text{доп}}) - F(x^n)).$$

Для этого алгоритма составляем процедуру **NIUTON**.

Procedure NIUTON(Var n:Integer;Var x:Real);

Var

Dx,L,Eras:Real;

Begin

n:=0;

Repeat

```

L:=F(x);
x:=x+Edop;
Dx:=L*Edop/(F(x)-L);
x:=x-Dx-Edop;
Eras:=ABS(Dx);
n:=n+1;
Until Eras<Edop;
End;

```

6.3. Создание программы расчета индукции и напряженности в зубцах электрической машины

Якорные обмотки электрических машин располагаются в пазах зубчатых якорей. Пазы обычно имеют прямоугольную форму, зубцы – трапециевидальную. Если магнитная система ненасыщенна, что соответствует примерно условию $B_z < 1,8$ Тл, то весь поток зубцового деления проходит через зубец, так как магнитная проводимость зубца при этом примерно в 1000 раз больше проводимости паза. При насыщении зубца часть потока ответвляется в паз. В этом случае индукцию в зубце можно найти по соотношению:

$$B_z = B_{z\text{рас}} - \mu_0 \cdot k_z \cdot H_z(B_z). \quad (6.4)$$

Здесь $B_{z\text{рас}} = \Phi_t / b_z$ – расчетная индукция в зубце, равная отношению потока зубцового деления на ширину зубца; $k_z = b_n / b_z$ – зубцовый коэффициент, равный отношению ширины паза к ширине зубца в данном сечении.

Уравнение (6.4) является нелинейным уравнением, так как зависимость $H_z(B_z)$ – это кривая намагничивания стали и она нелинейная. Для решения этого уравнения можно применить либо метод простых итераций, либо метод Ньютона. Составим программу, в которой будут использованы оба метода. Программа включает следующие структуры:

1. Функцию для кусочно-линейной интерполяции с постоянным шагом для кривой намагничивания;
2. Функцию для процедуры метода простых итераций, она составляется по выражению $f(B_z) = B_{z\text{рас}} - \mu_0 \cdot k_z \cdot H_z(B_z)$;

3. Функцию для процедуры метода Ньютона, она составляется по выражению $F(B_z) = B_z - B_{z\text{рас}} + \mu_0 \cdot k_z \cdot H_z(B_z)$;

4. Процедуру *NIUTON*;

5. Процедуру *PRIT*;

6. Головной модуль.

Далее приводится программа *ZUBEC* с комментариями.

Program ZUBEC;

Uses Crt;

Const

Db:Real=0.1; B0:Real=1.8; MO:Real=12.56E-7;

H:Array[0..5] of Real=(12500,25000,40000, 75000,130000,2300000);

Var

Kz,Br,x0,x,Edop:Real;

n:Integer;

Function HZ(x:Real):Real; { функция для кусочно-линейной интерполяции кривой намагничивания }

Var

i:Integer;

Begin

i:=TRUNC((x-B0)/Db);

HZ:=H[I]+(H[I+1]-H[I])*(x-B0-i*Db)/Db;

End;

Function F(x:Real):Real; { функция для метода простых итераций }

Begin

F:=Br-MO*Kz*HZ(x);

End;

Function F1(x:Real):Real; { функция для метода Ньютона }

Begin

F1:=x- Br+MO*Kz*HZ(x);

End;

Procedure NIUTON { из предыдущего подраздела }

Procedure PRIT { из предыдущего подраздела }

```

Begin { головной модуль }
  ClrScr;
  Writeln('введите Edop, X0,Kz,Br');
  Readln(Edop,X0,Kz,Br);
  x:=x0; NIUTON(n,x);
  Writeln('метод Ньютона','x=',x:7:5,' HZ(x)',HZ(x):10:2,' n=',n:5);
  x:=x0; PRIT(n,x);
  Writeln('метод простых итераций', 'x=',x:7:5,' HZ(x)=' ,HZ(x):10:2,'
n=',n:5);
End.

```

6.4. Содержание лабораторной работы

Для усвоения теоретического материала лекции выполните следующие задания:

1. По приведенной в подразделе 6.3 программе проведите расчеты для нескольких значений $X0$ и Br в диапазоне от 1,8 Тл до 2,3 Тл. Результаты расчета занесите в табл. 6.1.

Таблица 6.1 – Результаты расчета

x^0	$B_{зрас}$	Метод Ньютона			Метод простых итераций		
		n	x	H_z	n	x	H_z

Проанализируйте полученные результаты. Сравните методы по скорости сходимости.

2. Замените кусочно-линейную интерполяцию интерполяцией сплайнами. Внесите необходимые изменения в программу и повторите расчеты по пункту 1. Сравните результаты расчета. Сделайте выводы.

Контрольные вопросы

1. Запишите алгоритм метода простых итераций.

2. Назовите условие сходимости метода простых итераций.
3. Запишите алгоритм метода Ньютона.
4. Изобразите графическую иллюстрацию метода Ньютона.
5. Опишите физику процесса прохождения магнитного потока в зубцовой зоне электрической машины.
6. Какое физическое явление приводит к необходимости решать нелинейное уравнение для определения индукции в зубце?
7. Как отличаются формы записи для функции в методе простых итераций и в методе Ньютона?
8. Опишите структуру программы для расчета индукции в зубце электрической машины.

ЛЕКЦИЯ 7

МЕТОДЫ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ И ИХ ПРИМЕНЕНИЕ ДЛЯ РАСЧЕТА МАГНИТОДВИЖУЩЕЙ СИЛЫ ЗУБЦА ЭЛЕКТРИЧЕСКОЙ МАШИНЫ

Классификация методов численного интегрирования.

Методы прямоугольников. Алгоритмы. Процедуры.

Метод трапеций. Алгоритм. Процедура.

Метод Симпсона. Алгоритм. Процедура.

Определение магнитодвижущей силы зубца электрической машины.

Составление программы для расчета МДС зубца.

Содержание лабораторной работы.

7.1. Классификация методов численного интегрирования

Численные методы интегрирования находят применение в тех случаях, когда отсутствуют аналитические выражения для первообразной функции.

Суть численных методов заключается в том, что подынтегральную функцию аппроксимируют другими функциями, для которых существует аналитическое выражение для первообразной. Методы отличаются видом аппроксимирующей функции.

$$I = \int_{x_0}^{x^n} f(x) \cdot dx = \int_{x_0}^{x^n} \varphi(x) \cdot dx + R. \quad (7.1)$$

В выражении (7.1) $f(x)$ - реальная функция, $\varphi(x)$ – аппроксимирующая функция, R – погрешность определения интеграла численным методом.

Все существующие методы численного интегрирования можно разбить на 4 группы:

1. Методы полиномиальной аппроксимации Ньютона-Котеса, в которых подынтегральная функция аппроксимируется полиномами различных степеней;

2. Методы с применением сплайнов для аппроксимации функции;
3. Методы наивысшей точности Гаусса-Кристофеля, в которых применяются неравноотстоящие узлы и алгоритм расположения узлов определяется по критерию наименьшей погрешности;
4. Метод Монте-Карло, в котором расположение узлов определяется с помощью датчика случайных чисел. Метод эффективен для определения интегралов большой кратности.

С ростом числа разбиений интервала интегрирования n погрешность R в формуле (7.1) сначала уменьшается за счет более точной аппроксимации, а затем при значительном увеличении n погрешность возрастает, так как начинает сказываться влияние погрешностей округления.

7.2. Методы прямоугольников. Алгоритмы. Процедуры

В методе прямоугольников в качестве аппроксимирующей функции используется полином нулевого порядка P_0 , т.е. const . В качестве const можно принять любое значение функции на интервале, однако находят применение 3 метода прямоугольников для различных способов задания const :

1. **Метод средних прямоугольников** – const равна значению функции в средней точке интервала;
2. **Метод левых прямоугольников** – const равна значению функции в крайней левой точке интервала;
3. **Метод правых прямоугольников** – const равна значению функции в крайней правой точке интервала.

Рис. 7.1 иллюстрирует аппроксимацию функции перечисленными способами.

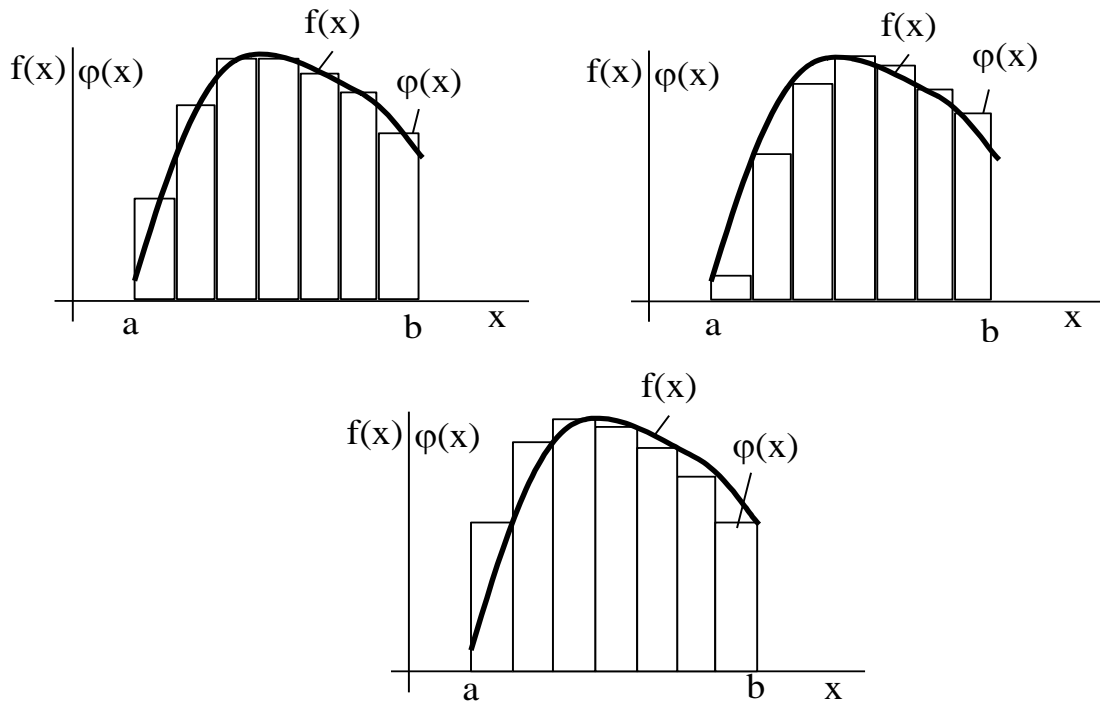


Рисунок 7.1 – Иллюстрация к методам средних, левых и правых прямоугольников

Далее приводим **алгоритмы** для определения интеграла на участке разбиения и для определения интеграла на всем интервале задания функции.

Метод средних прямоугольников

$$I_i = \int_{x_i}^{x_i+h} f(\bar{x}_i) \cdot dx = f(\bar{x}_i) \cdot h; \quad h = (b - a) / n;$$

$$I = \sum_{i=1}^n I_i = (f_{0.5} + f_{1.5} + \dots + f_{i-0.5} + \dots + f_{n-0.5}) \cdot h.$$

Метод левых прямоугольников

$$I_i = \int_{x_i}^{x_i+h} f(x_i) \cdot dx = f(x_i) \cdot h; \quad h = (b - a) / n;$$

$$I = \sum_{i=1}^n I_i = (f_0 + f_1 + \dots + f_{i-1} + \dots + f_{n-1}) \cdot h.$$

Метод правых прямоугольников

$$I_i = \int_{x_i}^{x_i+h} f(x_i + h) \cdot dx = f(x_i + h) \cdot h; \quad h = (b - a) / n;$$

$$I = \sum_{i=1}^n I_i = (f_1 + f_2 + \dots + f_i + \dots + f_n) \cdot h.$$

По приведенному алгоритму составим процедуру для метода средних прямоугольников.

Procedure Srednie(Var i:Real);

Var

h:Real;

j:Integer;

Begin

i:=0; { Обнуляем ячейку для накопления суммы }

h:=(b-a)/n; { вычисляем шаг интегрирования }

x:=a+h/2; { вычисляем среднее значение x на 1 участке }

For j:=1 to n do

Begin

i:=i+f(x); { накапливаем сумму чисел в скобках }

x:=x+h; { определяем x на следующих участках }

End;

*i:=i*h; { вычисляем приближенное значение интеграла }*

End;

Для метода левых и правых прямоугольников вносим изменения в процедуру, изменяем значение x на первом участке:

$x = a$ – для метода левых прямоугольников;

$x = a + h$ – для метода правых прямоугольников.

7.3. Метод трапеций. Алгоритм. Процедура

В методе трапеций в качестве аппроксимирующей функции применяется полином первого порядка P_1 , т.е. прямая линия. Из бесконечного мно-

жества возможных прямых обычно выбирают ту, которая проходит через узловые точки (смотрите рис. 7.2).

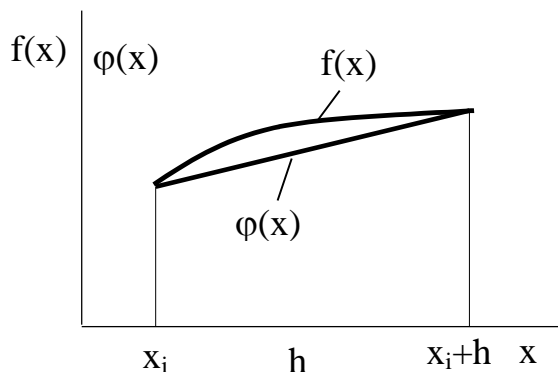


Рисунок 7.2 – Иллюстрация к методу трапеций

Запишем **алгоритмы** для определения приближенного значения интеграла на участке и на всем интервале задания функции.

$$I_i = \int_{x_i}^{x_i+h} \varphi(x) \cdot dx = h \cdot (f(x_i) + f(x_i + h)) / 2;$$

$$I = \sum_{i=1}^n I_i = h \cdot (f_0 / 2 + f_1 + \dots + f_i + \dots + f_n / 2);$$

$$h = (b - a) / n.$$

В соответствии с описанным алгоритмом составим процедуру.

Procedure trapecia(**var** i:**Real**);

Var

h:**Real**;

Begin

i:=(f(a)+f(b))/2;

h:=(b-a)/n;

For j:=1 **to** n-1 **do** i:=i+f(a+j*h);

i:=i*h;

End;

7.4. Метод Симпсона. Алгоритм. Процедура

В методе Симпсона в качестве аппроксимирующей функции используется полином 2-го порядка $P_2(x)$, который описывает параболу, проходящую через три соседних узла (рис. 7.3).

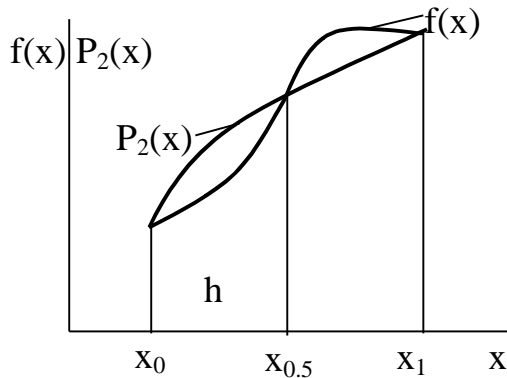


Рисунок 7.3 – Иллюстрация к методу Симпсона

Запишем **алгоритмы** приближенного вычисления интегралов для двух участков и для всего интервала изменения функции

$$I_i = \int_{x_0}^{x_1} P_2(x) \cdot dx = (f_0 + 4 \cdot f_{0.5} + f_1) \cdot h / 3;$$

$$I = (2 \cdot (f_0 / 2 + 2 \cdot f_{0.5} + f_1 + 2 \cdot f_{1.5} + \dots + f_n) - f_n) \cdot h / 3.$$

Для данного алгоритма составим процедуру.

```

Procedure Simpson(Var i:Real);
  Var
    h:Real;
    j:integer;
  Begin
    h:=(b-a)/(2*n); i:=f(a)/2; x:=a;
    For j:=1 to n do
      Begin
        x:=x+h; i:=i+2*f(x);
        x:=x+h; i:=i+f(x);
      End;
      i:=(2*i- f(x))*h/3;
    End;
  
```

7.5. Определение магнитодвижущей силы зубца электрической машины

В трапециидальных зубцах электрической машины магнитное поле является неоднородным, так как по высоте зубца (вдоль оси x) изменяется ширина зубца b_z , индукция и напряженность магнитного поля (рис. 7.4).

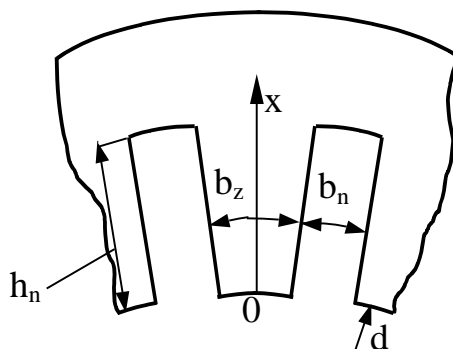


Рисунок 7.4 – Зубцовая зона электрической машины

На рис. 7.4 представлен элемент сердечника статора электрической машины переменного тока с внутренним диаметром d , шириной паза b_n и высотой паза h_n . (Высота зубца равна высоте паза).

Магнитодвижущая сила зубца в соответствии с законом полного тока определяется интегрированием напряженности зубца по его высоте:

$$F_z = \int_0^{h_n} H_{zx} \cdot dx.$$

Напряженность H_{zx} находим с помощью кривой намагничивания по значению индукции в данном сечении зубца B_{zx} . Индукцию B_{zx} определяем по следующему алгоритму:

1. Находим расчетное значение индукции в зубце при условии, что весь поток зубцового деления проходит через зубец по формуле $B_{zr} = \Phi_{tz} / b_{zx}$;
2. Если $B_{zr} < 1,8$ Тл, то действительное значение индукции в зубце равно расчетному значению индукции;
3. Если $B_{zr} > 1,8$ Тл, то реальное значение индукции в зубце B_{zx} находим из уравнения $B_{zx} = B_{zr} - \mu_0 \cdot k_{zx} \cdot H_{zx}(B_{zx})$ с помощью метода простых

итераций. В формуле $k_{zx} = b_n / b_{zx}$ – зубцовый коэффициент ; $b_{zx} = \pi(d + 2 \cdot x) / z_n - b_n$; z_n – число пазов статора, равное числу зубцов.

Исходные данные для расчета: высота и ширина паза – h_n, b_n , внутренний диаметр статора (диаметр расточки статора) – d , число пазов – z_n , поток зубцового деления на единицу длины машины – Φ_{tz} , кривая намагничивания стали, из которой изготовлен статор, допустимая погрешность определения индукции в зубце методом Ньютона – $edop$, число разбиений интервала интегрирования – n .

7.6. Составление программы для расчета МДС зубца

Составим программу в соответствии с описанным в подразделе 7.5 алгоритмом. Описательная часть программы содержит:

1. Описание констант (шаг таблицы для кривой намагничивания db , начальное значение аргумента в таблице bo , массив значений напряженности h в узлах таблицы и значение магнитной проницаемости воздуха $m0$);
2. Описание глобальных переменных (идентификаторы исходных данных $d, z_n, h_n, b_n, ft, n, edop$; идентификаторы пределов интегрирования a, b и результата интегрирования i ; bz, kz, br для метода простых итераций и координата x по высоте зубца);
3. Функцию $hz(bz)$ для интерполирования;
4. Функцию $f(x)$ для метода простых итераций;
5. Процедуру $Prit$;
6. Функцию $f3(x)$ для определения подынтегрального выражения;
7. Процедуру $Simpson$ для вычисления интеграла и определения МДС зубца.

В реализационной части программы приведены операторы ввода исходных данных, обращение к процедуре интегрирования *Simpson* и оператор вывода результата на печать.

Далее приводится программа с комментариями.

Program MDSZUB;

Uses Crt;

{ *описательная часть программы* }

Const

db:**Real**=0.1;

m0:**Real**=12.56e-7;

b0:**Real**=1.2;

h:**Array**[0..11] of **Real**=

(400,550,1000,1600,3400,7700,13400,19400,38800,65500,144000,224000
);

Var

d,zn,bn,hn,ft,edop,i,a,b,bz,kz,br,x:**Real**;

j,n:**Integer**;

Function hz(bz:**Real**):**Real**;

Var

i:**Integer**;

Begin

i:=trunc((bz-b0)/db);

hz:=h[i]+(h[i+1]-h[i])*(bz-b0-db*i)/db;

End;

Function f(x:**Real**):**Real**;*{ переписываем из программы преды-*
дущей лекции }

Procedure Prit(**Var** x:**Real**);*{ переписываем из программы*
предыдущей лекции и удаляем расчет числа итераций n }

Function f3(x:**Real**):**Real**;

Var

bzx:**Real**;

Begin

bzx:=pi*(d+2*x)/zn-bn; br:=ft/bzx;

if br<1.8 **Then**

begin

bz:=br; f3:=hz(bz)

end


```

        Else

Begin
    kz:=bn/bzx; bz:=br;
    Niuton(bz); f3:=hz(bz);
End;
End;
Procedure Simpson(Var i:Real);{ переписываем из подраздела
7.4}
{ реализационная часть программы}
Begin
    Writeln('введите edop,d,bn,hn,ft,zn,n');
    Readln(edop,d,bn,hn,ft,zn,n);
    a:=0; b:=hn;
    Simpson(i);
    Writeln('mds=',i:10:3);
End.

```

7.7. Содержание лабораторной работы

Целью лабораторной работы является усвоение знаний в области программирования, численных методов и электромеханики. Для достижения этой цели выполните следующее:

1. Переделайте описательную часть программы таким образом, чтобы описание констант, переменных, функций и процедур располагалось внутри тех функций и процедур, в которых они используются. Проверьте правильность новой программы путем сравнения результатов расчета по новой программе и по исходной. Исходные данные для расчета: $edop = 0,001$, $d = 0,9$, $bn = 0,0125$, $hn = 0,074$, $ft = 0,038$, $zn = 90$, $n = 8$. Результат расчета: МДС = 961 А.

2. Выполните расчеты по исходной программе для нескольких значений числа разбиений n . Замените процедуру *Simpson* процедурами *Trapezia*,

Srednie, Levie, Pravie поочередно, проведите расчеты и занесите результаты в сводную табл. 7.1.

Таблица 7.1.– Результаты расчета

Число разбиений n	Метод интегрирования				
	Симпсона	Трапеций	Средних прямоугольников	Левых прямоугольников	Правых прямоугольников
2					
4					
8					
16					
32					
64					

3. Проанализируйте результаты и сделайте выводы о методах интегрирования.

4. Выполните расчеты по исходной программе для различных значений потока ft , результаты расчета занесите в табл. 7.2.

Таблица 7.2.– Результаты расчета для разных значений потока ft

ft	0,034	0,036	0,038	0,04	0,042
МДС					

Сделайте выводы по результатам таблицы о влиянии насыщения на величину МДС зубца. Постройте график зависимости МДС от потока зубцового деления.

Контрольные вопросы

1. Запишите алгоритм метода прямоугольников (левых, правых и средних).
2. Запишите алгоритм метода трапеций.
3. Запишите алгоритм метода Симпсона.
4. Опишите физику задачи для расчета магнитодвижущей силы зубца электрической машины.
5. Опишите структуру программы для расчета МДС зубца электрической машины.
6. По результатам проведения лабораторной работы проведите сравнение методов интегрирования по критерию необходимого числа разбиений для обеспечения заданной точности.
7. Опишите алгоритм определения напряженности в зубце и его реализацию в функции $F3$.

ЛЕКЦИЯ 8

ОЦЕНКА ПОГРЕШНОСТИ ИНТЕГРИРОВАНИЯ И ИНТЕГРИРОВАНИЕ С ЗАДАННОЙ ТОЧНОСТЬЮ

Априорная оценка погрешности.

Апостериорная оценка погрешности. Формулы Рунге.

Создание процедуры определения уточненного значения интеграла с помощью формул Рунге.

Описание задачи расчета электромагнитного момента, магнитной энергии и коэнергии.

Создание программы для определения электромагнитного момента вентильно-индукторной машины.

Содержание лабораторной работы.

8.1. Априорная оценка погрешности

Получим формулу для оценки априорной погрешности для метода средних прямоугольников. Интеграл на участке $[x_i, x_i+h]$ по этому методу равен

$$I_i = \int_{x_i}^{x_i+h} f(\bar{x}_i) \cdot dx + R_{oi} = f(\bar{x}_i) \cdot h + R_{oi}. \quad (8.1)$$

Здесь R_{oi} – главный член погрешности на интервале. Для его определения раскладываем функцию $f(x)$ на участке в ряд Тейлора в окрестности точки \bar{x}_i

$$f(x) = f(\bar{x}_i) + \frac{f'(x) \cdot (x - \bar{x}_i)}{1!} + \frac{f''(x) \cdot (x - \bar{x}_i)^2}{2!} + \dots. \quad (8.2)$$

Запишем интеграл от функции $f(x)$ на участке

$$I_i = \int_{x_i}^{x_i+h} f(x) \cdot dx = f(\bar{x}_i) \cdot h + \frac{f'(\bar{x}_i) \cdot (x - \bar{x}_i)^2}{2} \Big|_{x_i}^{x_i+h} + \frac{f''(\bar{x}_i) \cdot (x - \bar{x}_i)^3}{6} \Big|_{x_i}^{x_i+h} + \dots.$$

После подстановки пределов получаем

$$I_i = f(\bar{x}_i) \cdot h + \frac{f''(\bar{x}_i) \cdot h^3}{24} + \dots \quad (8.3)$$

Пренебрегая слагаемыми с производными более высокого порядка из сравнения (8.1) и (8.3) получаем выражение для главного члена погрешности на участке

$$R_{oi} = \frac{f''(\bar{x}_i) \cdot h^3}{24}.$$

Определим главный член погрешности на всем интервале как сумму погрешностей на n участках

$$R_o = \sum_{i=1}^n R_{oi} = \frac{h^2}{24} \cdot \sum_{i=1}^n h \cdot f''(\bar{x}_i) = \frac{h^2}{24} \cdot \int_a^b f''(x) \cdot dx. \quad (8.4)$$

Степень шага в формуле для главного члена погрешности характеризует порядок метода. Метод средних прямоугольников имеет второй порядок. Прделав аналогичные вычисления, получим формулы для главных членов погрешности для других методов, рассмотренных в предыдущей лекции.

Методы левых и правых прямоугольников имеют первый порядок –

$$R_o = \frac{h}{2} \cdot \int_a^b f'(x) \cdot dx.$$

Метод трапеций имеет второй порядок –

$$R_o = -\frac{h^2}{12} \cdot \int_a^b f''(x) \cdot dx.$$

Метод Симпсона имеет четвертый порядок –

$$R_o = -\frac{h^4}{180} \cdot \int_a^b f''''(x) \cdot dx.$$

8.2. Апостериорная оценка погрешности.

Формулы Рунге

Формулы Рунге позволяют определить по результатам численного интегрирования с двумя различными значениями шага h следующие величины:

1. Главный член погрешности R_0 (первая формула Рунге);
2. Уточненное значение интеграла (вторая формула Рунге).

Выведем эти формулы. Представим точное значение интеграла I как сумму приближенного значения интеграла с шагом $h - I_h$ и главного члена погрешности R_0 :

$$I = I_h + R_0. \quad (8.5)$$

Запишем главный член погрешности в следующем виде:

$$R_0 = A h^p. \quad (8.6)$$

где p – порядок метода. После подстановки (8.6) в (8.5) получим

$$I = I_h + A \cdot h^p. \quad (8.7)$$

При интегрировании с шагом $k \cdot h$ аналогичные соотношения имеют вид:

$$I = I_{kh} + A \cdot (k \cdot h)^p = I_{kh} + A \cdot h^p \cdot k^p = I_{kh} + R_0 \cdot k^p. \quad (8.8)$$

Приравниваем правые части уравнений (8.5) и (8.8)

$$I_h + R_0 = I_{kh} + R_0 \cdot k^p.$$

Отсюда получаем **первую формулу Рунге**

$$R_0 = \frac{I_h - I_{kh}}{k^p - 1}. \quad (8.9)$$

Введем обозначение $k_r = k^p - 1$. Рассчитаем значения k_r для рассматриваемых методов интегрирования при $k = 2$ и занесем результаты в табл. 8.1.

Таблица 8.1– Значения коэффициента k_r

Метод	Средних прямоугольников, трапеций	Симпсона	Левых и правых прямоугольников
Коэффициент k_r	3	15	1

Уточненное значение интеграла определяем по **второй формуле Рунге**

$$I_{\text{уточ}} = I_h + R_0. \quad (8.10)$$

8.3. Создание процедуры определения уточненного значения интеграла с помощью формул Рунге

Алгоритм процедуры состоит из следующих этапов:

1. Задаемся минимальным числом разбиений интервала интегрирования $n = 2$ и определяем приближенное значение интеграла I одним из рассмотренных методов;

2. В цикле присваиваем значение интеграла I другой переменной I_1 , увеличиваем число разбиений в 2 раза, определяем новое значение интеграла I и рассчитываем главный член погрешности R_o по первой формуле Рунге. Условие окончания цикла $R_o < \varepsilon_{\text{доп}}$, где $\varepsilon_{\text{доп}}$ – допустимая погрешность интегрирования;

3. Определяем уточненное значение интеграла по второй формуле Рунге.

Создадим процедуру *UTOCHNIT* в соответствии с приведенным алгоритмом.

Procedure UTOCHNIT(**Var** IU:**Real**);

Var

Ro, I1, Edop, Kr:**Real**;

Begin

Writeln('Edop,Kr');

Readln(Edop,Kr);

n:=2; SIMPSON(I);

Repeat

I1:=I; n:=n*2;

SIMPSON(I); Ro=(I-I1)/Kr;

Until ABS(Ro)<Edop;

IU:=I1+Ro;

End;

8.4. Описание задачи расчета электромагнитного момента, магнитной энергии и коэнергии

Рассмотрим задачу расчета электромагнитного момента для вентиляно-индукторных машин. Наиболее широко для определения момента в этих машинах применяют методы, основанные на энергетическом подходе. Запишем расчетные формулы для этих методов

$$M_{эм} = - \left. \frac{dW_M(\Psi, \Theta)}{d\Theta} \right|_{\Psi = \text{const}} ; \quad (8.11)$$

$$M_{эм} = \left. \frac{dW_{co}(i, \Theta)}{d\Theta} \right|_{i = \text{const}} . \quad (8.12)$$

В (8.11) и (8.12) приняты следующие обозначения: W_M – магнитная энергия, W_{co} – коэнергия, Θ – угол между полюсами статора и ротора (рис. 8.1), i – ток в катушке, расположенной на полюсе статора, Ψ – потокосцепление катушки. Магнитная энергия W_M и коэнергия W_{co} определяются из соотношений:

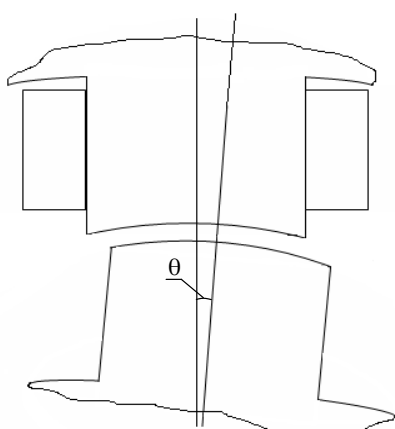


Рисунок 8.1 – Угол Θ

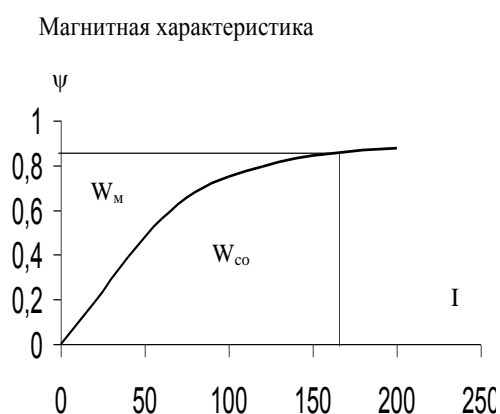


Рисунок 8.2 – Коэнергия и магнитная энергия

$$W_M = \int_0^{\Psi_k} i(\Psi, \Theta) \cdot d\Psi; \quad (8.13)$$

$$W_{co} = \int_0^{i_k} \Psi(i, \Theta) \cdot di. \quad (8.14)$$

Здесь $\Psi(i, \Theta)$ – магнитная характеристика, определенная экспериментально или с помощью расчетов магнитного поля одним из методов, например методом конечных элементов. Физический смысл магнитной энергии и коэнергии иллюстрирует рис. 8.2. Зависимость $\Psi(i, \Theta)$ – нелинейная зависимость, которую сложно аппроксимировать аналитическим выражением. Поэтому для определения интегралов в (8.13) и (8.14) применяют методы численного интегрирования с использованием таблично заданных зависимостей $\Psi(i, \Theta)$ -и $i(\Psi, \Theta)$, а электромагнитный момент определяется приближенно по выражениям

$$M_{эм} = - \left. \frac{W_M(\Psi_k, \Theta_2) - W_M(\Psi_k, \Theta_1)}{\Delta \Theta} \right|_{\Psi_k = \text{const}} ; \quad (8.15)$$

$$M_{эм} = \left. \frac{W_{co}(i_k, \Theta_2) - W_{co}(i_k, \Theta_1)}{\Delta \Theta} \right|_{i_k = \text{const}} . \quad (8.16)$$

Здесь $\Delta \Theta = \Theta_2 - \Theta_1$ – небольшое изменение угла Θ . Формулы (8.15) и (8.16) иллюстрируют рис. 8.3 и 8.4

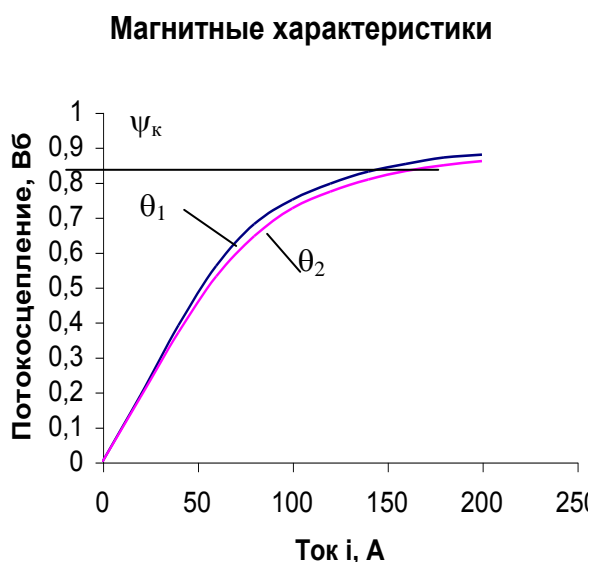


Рисунок 8.3 – К определению $M_{эм}$

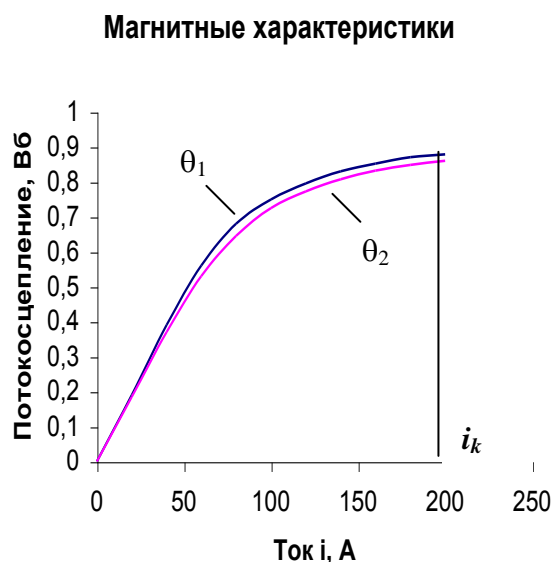


Рисунок 8.4 – К определению $M_{эм}$

На этих рисунках площади между кривыми характеризуют величину мгновенного значения момента $M_{эм}(\Psi_k, \Theta_1)$ и $M_{эм}(i_k, \Theta_1)$.

8.5. Создание программы для определения электромагнитного момента вентильно-индукторной машины

Составим программу для определения зависимости электромагнитного момента от тока. Электромагнитный момент будем рассчитывать по формуле (8.16) через изменение коэнергии.

Исходные данные для работы программы следующие:

1. 2 таблицы для зависимостей $\Psi(i, \Theta_1)$ и $\Psi(i, \Theta_2)$ с постоянным шагом по току;
2. Величина шага по току, начальное и конечное значение тока и число узлов в таблицах;
3. Допустимая погрешность интегрирования;
4. $\Delta\Theta = \Theta_2 - \Theta_1$.

Для определения промежуточных значений в таблице используем кусочно-линейную интерполяцию с постоянным шагом (функция *FUNCTION F1*). Для определения коэнергии по формуле (8.4) применим метод Симпсона с уточнением (процедуры *SIMPSON* и *UTOCHNIT*). Для расчета момента по формуле (8.16) составим процедуру *MOMENT1*.

Далее приведена программа, составленная в соответствии с описанным алгоритмом

Program MOMENT; {определение электромагнитного момента через изменение коэнергии}

Uses Crt;

Type

Vec=Array[0..10] of **Real**;

Const

PSI1:Vec=(0,0.185,0.37,0.53,0.645,0.725,0.77,0.81,0.832,0.85,0.86);

{значения потокосцеплений для $\Theta = \Theta_1$ }

PSI2:Vec=(0,0.19,0.39,0.56,0.68,0.75,0.795,0.83,0.852,0.87,0.878);

{значения потокосцеплений для $\Theta = \Theta_2$ }

x0=0; xk=200; dx=20; n1=10; {задание начального, конечного значения тока, шага по току и числа узлов в таблице}

```

Var
    edop,dtetta,mom,x,a,b,i,iu:Real;
    n,j,k:Integer;
    mom1,wco1,wco2,f:Vec;
Function F1(x:Real):Real;
Var
    i:Integer;
Begin
    i:=trunc(x/dx);
    F1:=F[i]+(F[i+1]-F[i])*(x-dx*i)/dx;
End;
Procedure SIMPSON(Var i:Real);{переписываем из раздела 7.4 преды-
дущей лекции}
Procedure UTOCHNIT(Var iu:Real);{переписываем из раздела 8.3 дан-
ной лекции}
Procedure MOMENT1(Var Mom:Real;k:Integer);
Begin
    F:=PSI1; UTOCHNIT(iu);
    wco1[k]:=iu;
    F:=PSI2; UTOCHNIT(iu);
    wco2[k]:=iu;
    mom :=(wco2[k]-wco1[k])/dtetta;
End;
Begin{Реализационная часть программы}
    ClrScr;
    Writeln('edop,dtetta');
    Readln(edop,dtetta);{вводим абсолютную допустимую погреш-
ность и приращение угла  $\Theta$  в градусах}
    dtetta:=dtetta*pi/180;{переводим угол в радианы}
    For k:=1 to n1 do
        Begin
            b:=(xk-0.001)*k/n1; a:=0;

```

```

MOMENT1(mom,k); mom1[k]:=mom;
Writeln('mom=',mom1[k]:7:2,'
wco1=',wco1[k]:7:2,' wco2=',wco2[k]:7:2);
End;
End.

```

8.6. Содержание лабораторной работы

Напоминаем, что практика в данном курсе преследует три цели:

1. Совершенствование навыков программирования;
2. Усвоение численных методов расчета;
3. Усвоение физики рассматриваемых задач.

Для достижения первой цели выполните следующее:

1. Составьте процедуру для построения графиков зависимостей момента и коээнергии от тока;
2. Внесите изменения в программу, которые позволят избежать пере-присвоения элементов массивов *PSI1*, *PSI2* элементам массива *F* в процедуре *MOMENT1*.

Для достижения второй цели проведите расчеты по программе для различных методов интегрирования, выведите на печать число разбиений n , которое обеспечивает заданную точность, и занесите результаты в табл. 8.1.

Таблица 8.1 – Результаты расчета

Метод	Левых пр-ков	Правых пр-ков	Средних пр-ков	Симпсона	Трапеций
n					

Для достижения третьей цели проведите расчеты по программе для других углов Θ (магнитные характеристики для этих углов должен предоставить преподаватель). Сделайте вывод о влиянии угла Θ на величину электромагнитного момента.

Контрольные вопросы

1. Назовите первую формулу Рунге.
2. Назовите вторую формулу Рунге.
3. Опишите алгоритм процедуры для интегрирования на заданную точность.
4. Назовите формулы для определения электромагнитного момента энергетическим методом.
5. Изобразите графическую иллюстрацию понятий „магнитная энергия” и „коэнергия”.
6. Запишите формулы для расчета магнитной энергии и коэнергии.
7. Запишите приближенную формулу для определения электромагнитного момента.
8. Опишите структуру программы для расчета электромагнитного момента вентильно-индукторного двигателя.

ЛЕКЦИЯ 9

ПРАКТИЧЕСКИЙ ГАРМОНИЧЕСКИЙ АНАЛИЗ

Термины и определения. Расчетные соотношения.

Составление программы для проведения гармонического анализа функций.

Примеры применения гармонического анализа в электромеханике.

Содержание лабораторной работы.

9.1. Термины и определения. Расчетные соотношения.

Практический гармонический анализ (ПГА) – это определение коэффициентов разложения функции в тригонометрический ряд Фурье методами численного интегрирования.

Запишем разложение функции $f(x)$ в ряд Фурье на интервале $[a, b]$

$$f(x) = a_0 + \sum_{k=1}^m a_k \cdot \cos\left(k \cdot \frac{2 \cdot \pi \cdot x}{b-a}\right) + \sum_{k=1}^m b_k \cdot \sin\left(k \cdot \frac{2 \cdot \pi \cdot x}{b-a}\right). \quad (9.1)$$

Здесь приняты следующие обозначения:

k – номер гармоники; m – число гармоник; a_0 , a_k , b_k – коэффициенты гармоник соответственно постоянной, косинусной и синусной составляющих. Коэффициенты определяются по соотношениям:

$$a_0 = \frac{1}{b-a} \int_a^b f(x) \cdot dx; \quad (9.2)$$

$$a_k = \frac{2}{b-a} \int_a^b f(x) \cdot \cos\left(k \cdot \frac{2 \cdot \pi \cdot x}{b-a}\right) \cdot dx; \quad (9.3)$$

$$b_k = \frac{2}{b-a} \int_a^b f(x) \cdot \sin\left(k \cdot \frac{2 \cdot \pi \cdot x}{b-a}\right) \cdot dx. \quad (9.4)$$

Если значения функции заданы в форме таблицы, то для определения интегралов применяют метод левых прямоугольников. Если имеется алгоритм по определению функции в любой точке интервала, может быть применен любой численный метод интегрирования. Для методов прямоуголь-

ников формулы (9.3) и (9.4) могут быть преобразованы. Покажем это на примере формулы (9.3). Заменим интеграл суммой

$$a_k = \frac{2}{b-a} \sum_{i=1}^n f(x_i) \cdot \cos(k \cdot \frac{2 \cdot \pi \cdot x_i}{b-a}) \cdot \Delta x. \quad (9.5)$$

$$\Delta x = \frac{b-a}{n}.$$

Подставим в формулу (9.5) и окончательно получим

$$a_k = \frac{2}{n} \sum_{i=1}^n f(x_i) \cdot \cos(k \cdot \frac{2 \cdot \pi \cdot x_i}{b-a}). \quad (9.6)$$

9.2. Составление программы для проведения гармонического анализа функций

Составим программу для определения коэффициентов ряда Фурье для любой функции для заданного числа гармоник по формулам (9.2), (9.3), (9.4) с проверкой ошибки замены функции конечным рядом и с графической иллюстрацией разложения функции на гармоники.

Программа будет состоять из следующих блоков:

1. *FUNCTION F(x)* для расчета значений функции в заданных точках;
2. *PROCEDURE SRPR* для расчета интегралов в формулах (9.3), (9.4).

Применяем для численного интегрирования метод средних прямоугольников;

3. *PROCEDURE KOEFGARM* для определения коэффициентов ряда Фурье по формулам (9.2), (9.3), (9.4);

4. *PROCEDURE MISTAKE* для определения максимальной ошибки при замене функции конечным рядом;

5. *PROCEDURE GRAFIKA* для построения следующих зависимостей: исходной кривой, кривой по выражению (9.1) и гармонических составляющих.

Для иллюстрации работы программы выбрана функция, которая описывается следующими соотношениями:

$$0 < x \leq 3 \quad f(x) = x;$$

$$\begin{aligned} 3 < x \leq 8 & \quad f(x) = 6 - x; \\ 8 < x \leq 10 & \quad f(x) = x - 10. \end{aligned}$$

График функции приведен на рис. 9.1.



Рисунок 9.1– График функции

Далее приводим описание программы с комментариями.

Program GARMONIK;

Uses

Crt,Graph;

Type

Vec=**Array**[1..20] of **Real**;

Var

i,k,m,n:**Integer**;

sc,ss,x,a,b,a0:**Real**;

ak,bk,ck:vec;{косинусные, синусные и результирующие коэффициенты гармоник}

y,y1:**Array**[1..2000] of **Real**;*{массивы для построения графиков исходной функции и функции по выражению (9.1)}*

y_{max},y_{min}:**Real**;

Function F(x:**Real**):**Real**;

Begin


```

If (x>=0) And (x<=3) Then F:=x;
      If (x>3) And (x<=8) Then F:=6-x;
      If (x>8) And (x<=10) Then F:=x-10;
End;

```

Procedure SRPR;

Var

 i:**Integer**;

 h:**Real**;

Begin

 h:=(b-a)/n;

 x:=a+h/2;

 ss:=0; sc:=0; ymax:=0; ymin:=0;

For i:=1 **To** n **Do**

Begin

 ss:=ss+f(x)*sin(k*2*pi*x/(b-a));

 sc:=sc+f(x)*cos(k*2*pi*x/(b-a));

 y[i]:=f(x);

If y[i]>ymax **Then** ymax:=y[i];

If y[i]<ymin **Then** ymin:=y[i];

 x:=x+h;

End;

 ss:=ss*h; sc:=sc*h;

End;

Procedure KOEFGARM;

Begin

Writeln('ВВОДИМ a b n m');

Readln(a,b,n,m);

 k:=0; SRPR;

 a0:=sc/(b-a);

Writeln('a0=',a0:10:4);

For k:=1 **To** m **Do**

Begin

```

        SRPR;
        ak[k]:=2*sc/(b-a); bk[k]:=2*ss/(b-a);
        ck[k]:=sqrt(sqr(ak[k])+sqr(bk[k]));
        Writeln('k=',k:3,' ak=',ak[k]:8:3,' bk=',bk[k]:8:3,
        ,ck='ck[k]:8:3);
    End;
End;
Procedure MISTAKE;
Var
    Erac,eracm:Real;
    i:Integer;
Begin
    Eracm:=0;
    For i:=1 To n Do
        Begin
            y1[i]:=a0;
            For k:=1 To m Do
                y1[i]:=y1[i]+ak[k]*cos(k*2*pi*i/n)+bk[k]*sin(k*2*pi*i/n);
                Erac:=ABS(y1[i]-y[i]);If erac>eracm
                    Then eracm:=erac;
            End;
        Writeln('eracm=',eracm:10:4);
    End;
Procedure GRAFIKA;
Var
    gd,gm,ycr:Integer;
    mx,my:Real;
Begin
    Gd:=detect;
    INITGRAPH(gd,gm,' ');
    my:=(getmaxy-40)/(ymax-ymin); mx:=getmaxx/n;
    ycr:=getmaxy-20-ymin;

```

```

Setbkcolor(3);
For i:=1 To n Do
    Begin
        Putpixel(round(i*mx),getmaxy-ycr,1);
        Putpixel(round(i*mx),getmaxy-ycr-round(y[i]*my),12);
        Putpixel(round(i*mx),getmaxy-ycr-round(y1[i]*my),13);
    End;
    For k:=1 To m Do
        For i:=1 To n Do
            Putpixel(round(i*mx),getmaxy-ycr-
            round(my*(a0+ak[k]*cos(k*2*pi*i/n)+
            bk[k]*sin(k*2*pi*i/n)),k+5);
        End;
Begin
    ClrScr;
    KOEFGARM; MISTAKE; GRAFIKA;
    Repeat Until Keypressed;
    Closegraph;
End.

```

9.3. Примеры применения гармонического анализа в электромеханике

Разложение в тригонометрический ряд Фурье различных физических величин – токов, напряжений, индукций, электродвижущих и магнитодвижущих сил – широко применяется в теории электрических машин. Перечислим некоторые из этих применений:

1. Разложение продольного и поперечного поля реакции якоря синхронной машины с целью определения первой гармоники;
2. Разложение проводимости воздушного зазора индукторной машины при использовании метода гармонических проводимостей;
3. Разложение кривых индукции на различных участках магнитной цепи вентильно-индукторного двигателя для расчета магнитных потерь.

4. Разложение кривой тока холостого хода трансформатора на гармоники.

Последняя задача 4 будет подробно рассмотрена в следующей лекции, остальные три могут быть использованы для практических заданий данного занятия.

9.4. Содержание лабораторной работы

Выполните следующие задания:

1. С помощью приведенной в подразделе 9.3. программы проведите расчеты для различного числа гармоник m . Значения максимальной погрешности занесите в табл. 9.1.

Таблица 9.1 – Значения максимальной погрешности в зависимости от числа гармоник m

m	5	10	15	20	25	30	35
$E_{\text{макс}}$							

По результатам табл. 9 и на основании графических иллюстраций сделайте вывод о достаточном числе гармоник для замены кривой гармоническим рядом.

2. Повторите п.1 для другой кривой, заданной руководителем.
3. Добавьте в программу вывод поясняющего текста к выводу результатов расчета.
4. Организуйте вывод результатов расчета во внешний файл.

Контрольные вопросы

1. Запишите выражение для тригонометрического ряда Фурье.
2. Запишите формулы для расчета коэффициентов ряда Фурье.
3. Опишите структуру программы для разложения произвольной функции в ряд Фурье с вычислением максимальной погрешности аппроксимации.
4. Как организован вывод на экран графиков зависимостей гармонических составляющих?
5. Опишите процедуру для определения максимальной погрешности аппроксимации.

ЛЕКЦИЯ 10

ОПРЕДЕЛЕНИЕ ГАРМОНИЧЕСКОГО СОСТАВА НАМАГНИЧИВАЮЩЕГО ТОКА ТРАНСФОРМАТОРА

Формулы для расчета коэффициентов ряда Фурье в частных случаях.

Форма намагничивающего тока трансформатора.

Расчет намагничивающего тока трехфазного трехстержневого трансформатора.

Создание программы для определения гармонического состава намагничивающего тока трансформатора.

Содержание лабораторной работы.

10.1. Формулы для расчета коэффициентов ряда Фурье в частных случаях

Формулы для расчета коэффициентов ряда Фурье можно представить в следующем виде:

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cdot \cos kx \cdot dx = \frac{1}{\pi} \int_0^{\pi} (f(x) + f(-x)) \cdot \cos kx \cdot dx;$$
$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cdot \sin kx \cdot dx = \frac{1}{\pi} \int_0^{\pi} (f(x) - f(-x)) \cdot \sin kx \cdot dx.$$

Если $f(x)$ – четная, т.е. $f(x) = f(-x)$, то

$$a_k = \frac{2}{\pi} \int_0^{\pi} f(x) \cdot \cos kx \cdot dx; b_k = 0.$$

Если $f(x)$ – нечетная, т.е. $f(x) = -f(-x)$, то

$$b_k = \frac{2}{\pi} \int_0^{\pi} f(x) \cdot \sin kx \cdot dx; a_k = 0.$$

Если $f(x) = -f(x + \pi)$, т.е. кривая, относящаяся к одной половине периода, является зеркальным изображением другой половины периода, то

$$a_{2k+1} = \frac{2}{\pi} \int_0^{\pi} f(x) \cdot \cos(2k+1)x \cdot dx; a_{2k} = 0;$$

$$b_{2k+1} = \frac{2}{\pi} \int_0^{\pi} f(x) \cdot \sin(2k+1)x \cdot dx; b_{2k} = 0.$$

10.2 Форма намагничивающего тока трансформатора

Намагничивающий ток трансформатора i_{μ} – это ток, который создает магнитный поток в сердечнике трансформатора. Для объяснения формы этого тока удобно воспользоваться законом Ома для магнитной цепи

$$i_{\mu} \cdot w_1 = \Phi \cdot R_{\mu}.$$

Здесь w_1 – число витков первичной обмотки; Φ – мгновенное значение потока в сердечнике, который изменяется во времени по синусоидальному закону; R_{μ} – магнитное сопротивление сердечника. Сердечник трансформатора изготавливается из ферромагнитного материала, который обладает свойством **насыщения**. Если сердечник трансформатора ненасыщен, магнитное сопротивление сердечника R_{μ} не зависит от величины потока, т.е. постоянно. В этом случае форма намагничивающего тока будет такой же, как и форма потока, т.е. синусоидальной. При насыщении сердечника его магнитное сопротивление при увеличении потока возрастает, при этом намагничивающий ток приобретает несинусоидальную форму (рис. 10.1).

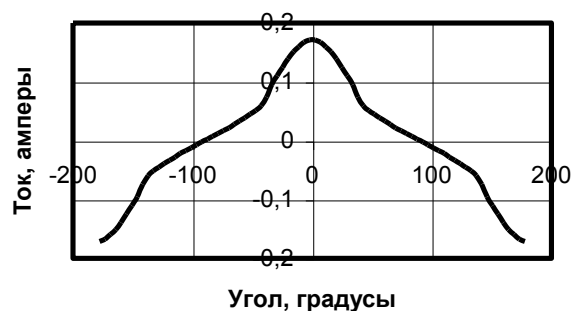


Рисунок 10.1. – Намагничивающий ток

Кривая на рис. 10.1 в соответствии с изложенным в п. 10.1 содержит только нечетные косинусные гармоники, т.е. разложение намагничивающего тока в ряд Фурье имеет вид

$$i_{\mu} = i_{\mu 1} \cdot \cos x + i_{\mu 3} \cdot \cos 3x + i_{\mu 5} \cdot \cos 5x + \dots \quad (10.1)$$

Здесь $i_{\mu 1} = a_1$, $i_{\mu 3} = a_3$, $i_{\mu 5} = a_5$.

$$a_{2k+1} = \frac{4}{\pi} \int_0^{\pi/2} f(x) \cdot \cos(2k+1)x \cdot dx; k = 0, 1, 2, \dots \quad (10.2)$$

10.3 Расчет намагничивающего тока трехфазного трехстержневого трансформатора

Для расчета намагничивающего тока трансформатора используют закон полного тока

$$\oint \vec{H} \cdot d\vec{l} = \sum i. \quad (10.3)$$

Здесь $\sum i$ – сумма токов, пронизывающих контур, по которому циркулирует вектор напряженности магнитного поля.

Обычно интегрирование проводят по средней силовой линии магнитной цепи. Ее разбивают на участки с однородным магнитным полем и интеграл заменяют суммой падений магнитного потенциала на участках, т.е.

$$\oint \vec{H} \cdot d\vec{l} \approx \sum H_{\text{уч}} \cdot l_{\text{уч}} = F = i_{\mu} \cdot W_1. \quad (10.4)$$

Здесь W_1 – число витков первичной обмотки трансформатора; F – магнитодвижущая сила (МДС), которая проводит поток по магнитной цепи. Для трехфазного трехстержневого магнитопровода отдельно определяют МДС для фазы А- F_A , для фазы В- F_B и для фазы С- F_C .

$F_A = F_C$, а $F_B < F_A$, так как для фазы В меньше длина средней силовой линии. Магнитный поток в каждой фазе проходит по следующим участкам: стержню, ярму и воздушным зазорам между пластинами.

Запишем выражения для МДС фаз в соответствии с формулой (10.4)

$$F_B = H_c \cdot h_c + H_a \cdot h_a + \frac{B_c}{\mu_0} \cdot \delta_z;$$

$$F_A = H_c \cdot h_c + H_a \cdot l_a + \frac{B_c}{\mu_0} \cdot \delta_z + 2 \cdot \frac{B_a}{\mu_0} \cdot \delta_z;$$

$$l_a = 2 \cdot l_c + d.$$

В этих формулах приняты следующие обозначения: B_c, B_a – индукции магнитного поля в стержне и ярме; H_c, H_a – напряженности магнитного поля в стержне и ярме; h_c, h_a – высота стержня и ярма; d_c – диаметр стержня; l_c – расстояние между осями стержней; δ_z – величина зазора между пластинами; μ_0 – магнитная проницаемость немагнитной среды.

Напряженности магнитного поля определяем для соответствующих индукций по кривым намагничивания для марки стали, из которой изготовлен магнитопровод. Индукции находим по соотношениям

$$B_c = \frac{\Phi}{P_c}; B_a = \frac{\Phi}{P_a}.$$

Здесь P_c, P_a – сечения стержня и ярма.

Поток Φ определяем по формуле

$$\Phi = \frac{U_1}{222 \cdot W_1} \cdot \cos x;$$

U_1 – эффективное значение первичного напряжения; $x = \omega t$.

После определения МДС фаз рассчитываем среднее значение МДС и величину намагничивающего тока по соотношениям

$$F_{cp} = \frac{2 \cdot F_A + F_B}{3}; i_\mu = \frac{F_{cp}}{W_1}.$$

10.4 Создание программы для определения гармонического состава намагничивающего тока

При разработке программы за основу принимаем универсальную программу, приведенную в лекции 9, и вносим в нее изменения. При этом возможны два варианта программ с соответствующими изменениями:

1. В программе лекции 9 заменяем описание функции. В новой программе функция предназначена для расчета намагничивающего тока. Составляем функцию по формулам подраздела 10.3. Расчет тока проводим на

периоде его изменения, поэтому в программе $a = 0$, $b = 2 \cdot \pi$. В головной модуль добавляем описание переменных, входящих в расчет намагничивающего тока, и ввод исходных данных для расчета тока. Рекомендуем выполнить эту работу самостоятельно.

2. Во втором варианте программы для определения коэффициентов ряда Фурье используем формулу (10.2). При этом расчет намагничивающего тока проводим на четверти периода, т.е. $a = 0$, $b = \pi / 2$. Рассчитываем только косинусные коэффициенты с нечетными номерами, поэтому вносим соответствующие изменения в процедуры *SRPR*, *KOEFGARM*, *MISTAKE*, *GRAFIKA*. Для построения графиков изменения функций на периоде рассчитываем значения функций на периоде по значениям функций на четверти периода с помощью процедуры *VSPGRAPH*. Кроме того, заменяем описание функции, вводим описание дополнительных переменных и ввод исходных данных для расчета тока, как в предыдущем варианте программы.

Приводим переделанную программу для второго варианта.

Program GARMONIK;

Uses

Crt,Graph;

Type

Vec=**Array**[1..20] of **Real**;

Var

i,k,m,n,n1,j:**Integer**;

d,pc,pa,hc,lc,u,w:**Real**;

sc,x,a,b:**Real**;

ak:vec;{косинусные коэффициенты гармоник}

y,y1,yg,y1g:**Array**[1..2000] of **Real**;{массивы для построения графиков исходной функции и функции по выражению (10.1) для четверти периода и для полного периода}

y_{max}:**Real**;

Function F(x:**Real**):**Real**;*{рассчитывает намагничивающий ток}*

Const

db=0.05; dz=0.5E-4; m0=12.56E-7;

```

    h:Array[0..32] Of Real=(0,6.5,13,19.5,26,32.5,39,45.5,
52,58.5,65,76,85,95,105,115,120,135,145,165,176,200,250,300,352,408,
461,540,620,760,900,1075,1250);
Var
    bc,ba,fa,fb:Real;
Function HT(BT:Real):Real;{кусочно-линейное интерполирование
кривой намагничивания}
    Var
        i:Integer;
    Begin
        i:=trunc(bt/db);
        ht:=h[i]+(h[i+1]-h[i])*(bt-i*db)/db;
    End;
Begin
    bc:=u*cos(x)/(222*w*pc);{индукция в стержне}
    ba:=bc*pc/pa;{индукция в ярме}
    fa:=2*ba*dz/m0+bc*dz/m0+ht(bc)*hc+ht(ba)*(2*lc+d);{МДС фазы A}
    fb:=bc*dz/m0+ht(bc)*hc+ht(ba)*d;{МДС фазы B}
    f:=(2*fa+fb)/(3*w);{намагничивающий ток}
End;
Procedure SRPR;{метод средних прямоугольников для численного ин-
тегрирования в формуле (2)}
    Var
        i:Integer;
        h:Real;
    Begin
        h:=(b-a)/n; x:=a+h/2;
        sc:=0; ymax:=0;
        For i:=1 To n Do
            Begin
                sc:=sc+f(x)*cos(k*x); y[i]:=f(x);
                If y[i]>ymax Then ymax:=y[i];
            End;
        End;
    End;

```

```

        x:=x+h;
    End;
    sc:=sc*h;
End;
Procedure KOEFGARM;{расчет коэффициентов ряда Фурье}
Begin
    Writeln('a b n m'); Readln(a,b,n,m);
    For j:=0 To m Do
        Begin
            k:=2*j+1;
            SRPR;
            ak[k]:=2*sc/(b-a);
            Writeln('k=',k:3,' ak=',ak[k]:8:3);
        End;
    End;

Procedure MISTAKE;{расчет ошибки замены реальной кривой намаг-
ничивающего тока кривой по уравнению (10.1) при данном числе учтенных
гармоник}
Var
    erac,eracm:Real;
    i:Integer;
Begin
    eracm:=0;
    For i:=1 To n Do
        Begin
            y1[i]:=0;
            For j:=0 To m Do
                Begin
                    k:=2*j+1;
                    Y1[i]:=y1[i]+ak[k]*cos(k*pi*i/(2*n));
                End;
            erac:=ABS(y1[i]-y[i]);
        End;
    End;

```

```

        If erac>eracm Then eracm:=erac;
    End;
    Writeln('eracm=',eracm:10:4);
End;
Procedure VSPGRAPH;{расчет координат функций для построения
графиков на периоде}
Begin
    For i:=1 To n Do Begin
        yg[i]:=y[n-i+1]; ylg[i]:=yl[n-i+1]; End;
    For i:=n+1 To 2*n Do Begin
        yg[i]:=y[i-n]; ylg[i]:=yl[i-n]; End;
    For i:=2*n+1 To 4*n Do Begin
        yg[i]:=-yg[i-2*n]; ylg[i]:=-ylg[i-2*n]; End;
    End;
Procedure GRAFIKA;{построение графиков исходной кривой, кривой
по ряду Фурье и гармонических составляющих}
Var
    Gd,gm,ycr:Integer;
    mx,my:Real;
Begin
    Gd:=detect;
    INITGRAPH(gd,gm,' ');
    my:=(getmaxy-40)/(2*ymax); mx:=getmaxx/n1;
    ycr:= getmaxy/2;
    Setbkcolor(3);
    For i:=1 To n1 Do Begin
        Putpixel(round(I*mx),getmaxy-ycr,1);{ось абсцисс}
        Putpixel(round(I*mx),getmaxy-ycr-round(yg[I]*my),12);{исходная
кривая}
        Putpixel(round(I*mx),getmaxy-ycr-round(ylg[I]*my),13);{ кривая по
ряду Фурье}
    End;

```

```

For j:=0 To m Do Begin
    k:=2*j+1;
    For i:=1 To n1 Do{построение гармоник}
        Putpixel(round(i*mx),getmaxy-ycr-
            round(my*(ak[k]*sin(k*2*pi*i/n1))),k+5);
    End;
End;
Begin{исполнительная часть головного модуля}
    ClrScr;
    Writeln('d,pc,pa,hc,lc,u,w');
    Readln(d,pc,pa,hc,lc,u,w);{ввод данных трансформатора}
    KOEFGARM;{вызов процедуры для расчета коэффициентов ряда
Фурье}
    MISTAKE;{вызов процедуры для расчета ошибки замены кривой ря-
дом Фурье}
    Readln;{необходимо для останова счета}
    n1:=n*4;
    VSPGRAPH;{вызов процедуры для расчета координат функций на
периоде}
    GRAFIKA;{вызов процедуры для построения графиков}
    Repeat Until Keypressed;
    Closegraph;
End.

```

10.5 Содержание лабораторной работы

Выполните следующие задания:

1. Для заданных руководителем данных трансформатора проведите расчеты по описанной в подразделе 10.4. программе для разного числа гармоник и занесите в табл. 10.1 значения максимальной погрешности.

Таблица 10.1– Значения максимальной погрешности

m	0	1	2	3	4	5	6
E_{max}							

По данным табл. 10.1 и на основании построенных графиков сделайте вывод о достаточном количестве гармоник для замены кривой намагничивающего тока рядом Фурье.

2. Для выбранного количества гармоник проведите расчеты при разных значениях напряжения больше и меньше заданного номинального в соответствии с табл. 10.2. Занесите значения коэффициентов ряда Фурье для 1-ой, 3-ей и 5-ой гармоник в табл. 10.2.

Таблица 10.2– Значения коэффициентов ряда Фурье для 1-ой, 3-ей и 5-ой гармоник

U	$1,1u_{ном}$	$u_{ном}$	$0,9u_{ном}$	$0,7u_{ном}$	$0,5u_{ном}$
a_1					
a_3					
a_5					

По данным табл. 10.2 сделайте вывод о влиянии насыщения на гармонический состав намагничивающего тока.

3. В соответствии с указаниями подраздела 10.4 создайте первый вариант программы и проведите ее отладку.

Контрольные вопросы

1. Запишите формулы для определения коэффициентов ряда Фурье для четных функций, нечетных функций и функций с зеркальным отображением.

2. Что такое „намагничивающий ток” и как влияет насыщение магнитной цепи на его форму (объясните с помощью закона Ома для магнитной цепи)?

3. Какой закон лежит в основе расчета намагничивающего тока?
4. Нарисуйте магнитную цепь трехфазного трансформатора и укажите на ней пути интегрирования по закону полного тока.
5. Какие высшие гармоники оказывают наибольшее влияние на форму намагничивающего тока при насыщении магнитной цепи.
6. Опишите структуру программы для определения гармонического состава намагничивающего тока трансформатора.
7. Как определяется достаточное количество гармоник для аппроксимации реальной кривой намагничивающего тока суммой гармоник?
8. Как влияет значение питающего напряжения на величину высших гармоник в кривой намагничивающего тока?

ЛЕКЦИЯ 11

ЧИСЛЕННОЕ РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Метод Эйлера. Алгоритм. Программа.

Метод Рунге-Кутты 2 порядка. Алгоритм. Программа.

Метод Рунге-Кутты 4 порядка. Алгоритм. Программа.

Содержание лабораторной работы.

11.1. Метод Эйлера. Алгоритм. Программа

Этот метод является простейшим и в то же время наименее точным численным методом решения дифференциальных уравнений 1 порядка. Запишем дифференциальное уравнение 1 порядка в форме Коши

$$\frac{dy(x)}{dx} = f(x, y). \quad (11.1)$$

$$\text{Начальное условие} - y(x_0) = y_0. \quad (11.2)$$

Разложим $y(x)$ в окрестности точки x_0 в ряд Тейлора

$$y(x) = y(x_0) + (x - x_0) \cdot y'(x_0) + \frac{(x - x_0)^2}{2} \cdot y''(x_0) + \dots \quad (11.3)$$

Обозначим: $x - x_0 = h$, тогда $x = x_0 + h$. Подставляем эти обозначения в формулу (11.3):

$$y(x_0 + h) = y(x_0) + h \cdot y'(x_0) + \frac{h^2}{2} \cdot y''(x_0) + \dots$$

При малом шаге h можно пренебречь третьим и всеми следующими слагаемыми; кроме того, из уравнения (11.1) следует, что $y'(x_0) = f(x_0, y_0)$. Получаем окончательно расчетную формулу (**алгоритм**) для метода Эйлера

$$y(x_0 + h) = y(x_0) + h \cdot f(x_0, y_0). \quad (11.4)$$

Локальная погрешность метода Эйлера имеет 2 порядок, а глобальная – 1-порядок. Так же, как при численном интегрировании, можно использовать для уточнения первую и вторую формулы Рунге

$$R_0 = \frac{y_h(x) - y_{kh}(x)}{k^p - 1}; y_h(x)_{\text{точ}} = y_h(x) + R_0.$$

Здесь p — порядок метода, т.е. 1 для метода Эйлера; $k=2$.

Формула Эйлера (11.4) обобщается для систем ОДУ, записанных в форме Коши (11.1)

$$y_i(x_0 + h) = y_0 + h \cdot f_i(x_0, y_{10}, \dots, y_{n0}); \quad i=1..n. \quad (11.5)$$

Приведем геометрическую интерпретацию метода Эйлера.

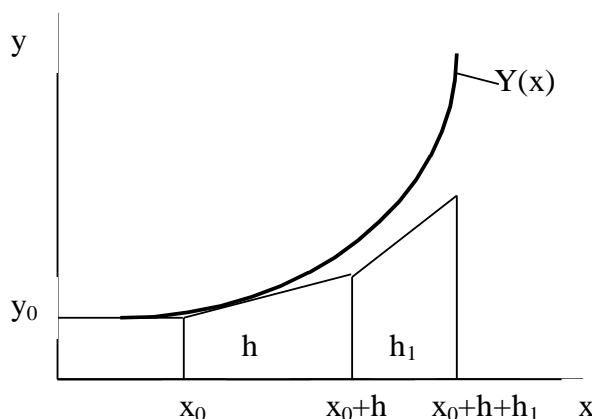


Рисунок 11.1— Иллюстрация к методу Эйлера

В точках x_0 , x_0+h , x_0+h+h_1 и т.д. проводим касательные к кривой $y(x)$, тем самым заменяем реальную кривую зависимостью, состоящей из отрезков прямых. Поэтому метод Эйлера называют методом ломаных. Очевидно, для того чтобы ломаная кривая совпадала с реальной кривой, шаг в методе Эйлера должен быть весьма малым. Обычно шаг принимают одинаковым на всем интервале интегрирования.

Составим **программу** для решения систем ОДУ методом Эйлера. В качестве примера для решения выберем уравнение Бесселя

$$y'' + y' / x + (1 - p^2 / x^2) \cdot y = 0, \quad (11.6)$$

где величина p является параметром уравнения, определяющим порядок цилиндрических функций, являющихся решением этого уравнения. Перейдем от дифференциального уравнения второго порядка к системе двух уравнений первого порядка в форме Коши. Обозначим неизвестную функ-

цию $y(x)$ через $y_1(x)$, а ее первую производную – $y_2(x)$, тогда получим систему ОДУ, эквивалентную уравнению (11.6)

$$\begin{aligned} y_1'(x) &= y_2(x), \\ y_2'(x) &= (p^2 / x^2 - 1) \cdot y_1(x) - y_2(x) / x. \end{aligned} \quad (11.7)$$

Для расчета правых частей уравнений (11.7), т.е. производных, составим процедуру *RP*, для реализации метода Эйлера (формула (11.5)) составим процедуру *EUL*. Далее приводим программу с комментариями.

Program Euler;

Uses Crt;

Type

Vec=Array[1..8] **Of Real**;

Var

p,x,xk,h:**Real**; {xk-конечное значение x}

Y,F:vec;

n:**Integer**;

Procedure RP(x:**Real**;Var Y,F:vec);{вычисление производных}

Begin

F[1]:=Y[2];

F[2]:=(sqr(p/x)-1)*Y[1]-Y[2]/x

End;

Procedure EUL(n:**Integer**);{метод Эйлера}

Var

i:**Integer**;

Begin

RP(x,Y,F);

For i:=1 **To** n **Do** Y[i]:=Y[i]+h*F[i];

End;

Begin{исполнительная часть головного модуля}

Writeln('n p x xk h Y[1] Y[2]');

Readln(n,p,x,xk,h,Y[1],Y[2]);

```

Repeat
  EUL(n);
  x:=x+h;
  Writeln('x=',x:5:2,' Y[1]=',Y[1]:10:6,' Y[2]=',Y[2]:10:6);
Until x>xk;
End;
End.

```

11.2. Метод Рунге-Кутты 2 порядка. Алгоритм. Программа

В этих методах учитывается член ряда Тейлора с h^2 . При этом локальная погрешность метода имеет третий порядок, глобальная – второй. Применяются методы с различными расчетными схемами. Рассмотрим *метод Рунге-Кутты 2 порядка с коррекцией по средней производной*.

Алгоритм метода имеет вид

$$y(x_0 + h) = y(x_0) + h \cdot (f_0 + f(x_0 + h, y_0 + h \cdot f_0))/2. \quad (11.8)$$

Этот алгоритм иллюстрирует рис. 11.2.

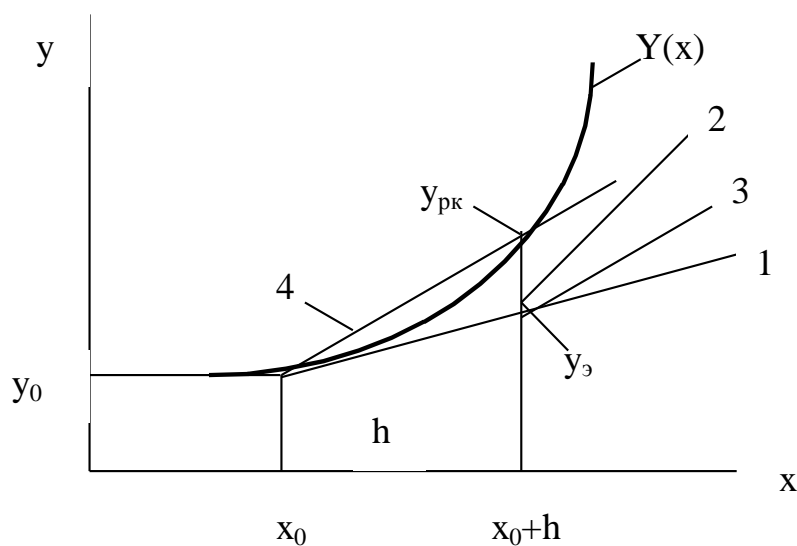


Рисунок 11.2 – Иллюстрация к методу Рунге-Кутты 2 порядка

Обозначения на рис. 11.2:

Прямые линии: 1 – касательная к кривой в точке $x=x_0$; 2 – прямая параллельная касательной к кривой в точке $x=x_0+h$; 3 – биссектриса угла между линиями 1 и 2; 4 – прямая параллельная линии 3; Y_9 – результат интегрирования по методу Эйлера; $Y_{рк}$ – результат интегрирования по методу Рунге-Кутта 2 порядка с коррекцией по средней производной.

Преобразуем формулу (11.8). По методу Эйлера

$$y_9 = y(x_0) + h \cdot f_0.$$

Отсюда

$$y(x_0) = y_9 - h \cdot f_0. \quad (11.9)$$

Подставляем (11.9) в формулу (11.8) и после преобразований окончательно получаем

$$y(x_0 + h) = y_9 + h \cdot (f(x_0 + h, y_0 + h \cdot f_0) - f_0) / 2; \quad (11.10)$$

$$y(x_0 + h) = y_9 + h \cdot (f(x_0 + h, y_9) - f_0) / 2. \quad (11.11)$$

Формула (11.11) является **алгоритмом** для составления программы. Составим **программу** для решения системы уравнений (11.11) методом Рунге-Кутта 2 порядка. За основу берем программу *EULER* и вносим в нее следующие изменения:

- 1) заменяем процедуру *EUL* процедурой *RK*;
- 2) добавляем описание вектора *F0:vec*;
- 3) в исполнительной части головного модуля исключаем оператор $x=x+h$ и заменяем обращение к процедуре.

Далее приводим процедуру *RK*.

Procedure RK(n:Integer);

Var

i:Integer;

Begin

RP(x,Y,f0);

For i:=1 **To** n **Do** Y[i]:=Y[i]+h*F0[i];

x:=x+h;

RP(x,Y,f);

For i:=1 To n Do Y[i]:=Y[i]+h*(F[i]-F0[i])/2;
End;

11.3. Методы Рунге-Кутты 4 порядка. Алгоритм. Программа

Для построения вычислительных схем методов Рунге-Кутты 4 порядка в тейлоровском разложении искомого решения $y(x)$ учитываются члены, содержащие степени шага h до четвертой включительно. После аппроксимации производных правой части ОДУ $f(x,y)$ получено семейство схем Рунге-Кутты 4 порядка. Наиболее используемой в вычислительной практике является следующая схема:

$$y(x_0 + h) = y_0 + (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4) / 6 + O(h^5). \quad (11.12)$$

Здесь

$$\begin{aligned} k_1 &= h \cdot f(x_0, y_0), \\ k_2 &= h \cdot f(x_0 + h/2, y_0 + k_1/2), \\ k_3 &= h \cdot f(x_0 + h/2, y_0 + k_2/2), \\ k_4 &= h \cdot f(x_0 + h, y_0 + k_3). \end{aligned}$$

Схема (11.12) на каждом шаге h требует вычисления правой части ОДУ в четырех точках. Локальная погрешность схемы имеет пятый порядок, глобальная – четвертый. Схема обобщается для систем ОДУ, записанных в форме Коши. Для удобства программной реализации, особенно в случае систем ОДУ, формулы (11.12) рекомендуется преобразовать к виду

$$y_i(x_0 + h) = y_{i0} + (q_{i1} + 2 \cdot q_{i2} + q_{i3} + q_{i4}) / 3, \quad (11.13)$$

где

$$\begin{aligned} q_{i1} &= h_2 \cdot f_i(x_0, y_{i0}), \quad h_2 = h/2, \\ q_{i2} &= h_2 \cdot f_i(x_0 + h_2, y_{i0} + q_{i1}), \\ q_{i3} &= h \cdot f_i(x_0 + h_2, y_{i0} + q_{i2}), \\ q_{i4} &= h_2 \cdot f_i(x_0 + h, y_{i0} + q_{i3}), \end{aligned}$$

$i = 1, 2, \dots, n$ – номер уравнения в системе ОДУ из n уравнений.

Уравнение (11.13) является **алгоритмом** для составления программы.

Составим программу для решения систем ОДУ методом Рунге-Кутты 4 порядка. За основу возьмем программу *EULER* и заменим в ней процедуру *EUL* процедурой *RK4*. Ниже приводится текст этой процедуры.

Procedure RK4(n:Integer;Var x,h:Real;Var Y:vec);

Var

i,j :Integer; h1,h2,q:Real;

Y0,Y1,F:vec;

Begin

h1:=0; h2:=h/2;

For i:=1 To n Do Begin Y0[i]:=Y[i]; Y1[i]:=Y[i]; **End;**

For j:=1 To 4 Do

Begin

RP(x+h1,Y,f);

If j=3 Then h1:=h **Else** h1:=h2;

For i:=1 To n Do

Begin

q:=h1*f[i];Y[i]:=Y0[i]+q;

If j=2 Then q:=2*q; Y1[i]:=Y1[i]+q/3.0;

End

End;

For i:=1 To n Do Y[i]:=Y1[i];

End;

11.4. Содержание лабораторной работы

Цели данной лабораторной работы следующие:

- 1) оценить свойства методов решения дифференциальных уравнений;
- 2) совершенствовать навыки программирования по работе с процедурами, вводом и выводом данных на экран и во внешние файлы.

Для достижения первой цели необходимо провести расчеты по приведенным программам на следующие исходные данные: $n=2$; $p=0$; $x=0,5$; $xk=1$; $Y[1]=0,9384698$; $Y[2]= - 0,2422685$ для двух значений шага $h=0,1$ и $h=0,05$. Сравнить значение $Y[1]$ при $xk=1$ с табличным для функции Бесселя нулевого порядка $Y_m[1] = 0,7651977$. Оценить точность методов путем вычисления погрешности расчета по формуле

$$\varepsilon = \frac{Y[1] - Y_m[1]}{Y_m[1]}.$$

Для достижения второй цели необходимо составить программу, которая обеспечит расчет по трем методам и вывод результатов расчета на экран и во внешний файл.

Контрольные вопросы

1. Запишите алгоритм метода Эйлера для решения дифференциальных уравнений.
2. Нарисуйте геометрическую интерпретацию метода Эйлера.
3. Запишите алгоритм метода Рунге-Кутты 2 порядка с коррекцией по средней производной.
4. Нарисуйте геометрическую интерпретацию метода Рунге-Кутты 2 порядка с коррекцией по средней производной.
5. Запишите алгоритм метода Рунге-Кутты 4 порядка.
6. Покажите, как дифференциальное уравнение 2 порядка преобразуется в систему уравнений первого порядка.
7. Сравните методы по результатам проведенной лабораторной работы.

ЛЕКЦИЯ 12

РАСЧЕТ ТОКА В КАТУШКЕ С ЖЕЛЕЗНЫМ СЕРДЕЧНИКОМ ПРИ ПИТАНИИ ПЕРЕМЕННЫМ НАПРЯЖЕНИЕМ ПРЯМОУГОЛЬНОЙ ФОРМЫ

(Пример применения численных методов решения ОДУ
в электротехнических задачах)

Описание задачи.

Описание программы на языке Паскаль.

Описание модели для программы SimuLink.

Содержание лабораторной работы.

12.1. Описание задачи

Для расчета тока в катушке необходимо решить дифференциальное уравнение для электрической цепи катушки

$$U = R \cdot i(\Psi) + \frac{d\Psi}{dt} \quad (12.1)$$

при заданном начальном условии

$$t=0; \psi = 0. \quad (12.2)$$

Исходные данные для расчета:

1. Амплитуда U_z и период T питающего напряжения;
2. Значение активного сопротивления катушки R ;
3. Таблица для магнитной характеристики $\psi(i)$;
4. Время окончания процесса X_k ;
5. Шаг интегрирования h .

Задачу будем решать двумя методами: 1) с помощью программы на языке *PASCAL*; 2) с помощью модели, составленной для программы *SIMULINK*, которая входит в пакет программ *MATHLAB*.

12.2. Описание программы на языке PASCAL

Составим *программу расчета тока в катушке* и построения графиков зависимостей тока, потокосцепления и напряжения от времени. Программа содержит следующие структурные элементы:

1. Функцию *FUNCTION F* для расчета правой части уравнения (12.1), записанного в форме Коши. Внутри функции *F* размещена функция *FUNCTION IT* для нахождения тока по магнитной характеристике с помощью кусочно-линейной интерполяции с переменным шагом;

2. Процедуру *PROCEDURE EULER* для решения дифференциального уравнения (12.1) методом Эйлера;

3. Процедуру *PROCEDURE GRAFIKA* для вывода на экран зависимостей тока, напряжения и потокосцепления от времени, т.е. $i(t)$, $u(t)$, $\psi(t)$.

Program COIL;

Uses

Crt, Graph;

Const

PSI: **Array** [0..10] **Of Real** =(0, 0.19, 0.39, 0.57, 0.68, 0.75, 0.8, 0.83, 0.855, 0.86, 0.878);

ITI: **Array** [0..10] **Of Real**=(0, 20, 40, 60, 80, 100, 120, 140, 160, 180, 200);

Var

x, y,xk, h,r,u,uz,T: **Real**;

U1, I1, X1, Y1: **Array** [0..1000] **Of Real**;

j,k: **Integer**;

Function F(x,y:**Real**):**Real**;*{функция для вычисления правой части уравнения (12.1) в форме Коши}*

Function IT(y:**Real**):**Real**;*{функция для кусочно-линейного*

Var *интерполирования}*

i: **Integer**;

Begin

i:=0;

Repeat i:=i+1 **Until** y<PSI[i];

IT:=ITI[i-1]+(ITI[i]-ITI[i-1])*(y-PSI[i-1])/((PSI[i]-PSI[i-1]));

End;

Begin

k:= Trunc(x/T);

```

If ( $x \geq k \cdot T$ ) And ( $x \leq (k \cdot T + T/2)$ ) Then  $u := uz$  Else  $u := -uz$ ;
 $F := u - IT(y) \cdot r$ ;
End;
Procedure EULER(Var  $h, x, y$ :Real);
Begin
     $y := y + h \cdot F(x, y)$ ;
End;
Procedure GRAFIKA;
Var
     $Gd, Gm, m$ :Integer;
     $mx, my, mu, mit$ :Real;
Begin
     $Gd := \text{Detect}$ ;
     $\text{Initgraph}(Gd, Gm, \text{'путь к драйверу'})$ ;
     $\text{Cleardevice}$ ;
     $mx := 640/xk$ ;  $my := 240/ymax$ ;  $it := 240/220$ ;  $mu := 220/uz$ ;
     $\text{Setcolor}(5)$ ;
     $\text{Line}(0, 240, 639, 240)$ ;
     $\text{Setcolor}(3)$ ;
    For  $m := 1$  To  $j-1$  Do {график для потокосцепления}
         $\text{Line}(\text{Round}(mx \cdot X1[m-1]), 240 - \text{Round}(my \cdot Y1[m-1]), \text{Round}(mx \cdot X1[m]), 240 - \text{Round}(my \cdot Y1[m]))$ ;
         $\text{Setcolor}(6)$ ;
    For  $m := 1$  To  $j-1$  Do {график для напряжения}
         $\text{Line}(\text{Round}(mx \cdot X1[m-1]), 240 - \text{Round}(mu \cdot U1[m-1]), \text{Round}(mx \cdot X1[m]), 240 - \text{Round}(mu \cdot U1[m]))$ ;
         $\text{Setcolor}(7)$ ;
    For  $m := 1$  To  $j-1$  Do {график для тока}
         $\text{Line}(\text{Round}(mx \cdot X1[m-1]), 240 - \text{Round}(mit \cdot I1[m-1]), \text{Round}(mx \cdot X1[m]), 240 - \text{Round}(mit \cdot I1[m]))$ ;
End;
Begin{головной модуль}

```

```

Clrscr;
T:=0.02;
x:=0; xk:=0.04; r:=0.01; uz:=83; j:=0; {ввод начальных зна-
h:=0.0001; y:=0; u:=uz; чений и исходных данных}
While x<xk Do
  Begin
    X1[j]:=x; {определяются элементы массивов для
    Y1[j]:=y; построения графиков}
    I1[j]:=IT(y); U1[j]:=u;
    EULER(h,x,y);
    j:=j+1;
    Writeln('x=',x:10:3,' y=',y:10:3,' IT=',IT(y):10:3);
    x:=x+h;
  End;
  Readln;
  GRAFIKA;
  Repeat Until Keypressed;
  Closegraph;
End.

```

12.3. Описание модели для программы SIMULINK

Пакет SimuLink позволяет осуществлять исследование (моделирование) поведения динамических нелинейных систем. Ввод характеристик этих систем производится в диалоговом режиме путем графической сборки схемы соединений элементарных стандартных звеньев. В результате такой сборки образуется модель исследуемой системы, которую будем называть S- моделью. В качестве кирпичиков для построения S-модели применяют модули (или блоки), хранящиеся в библиотеке SimuLink. Любая S-модель может иметь иерархическую структуру, т.е. состоять из моделей более низкого уровня (субмоделей). В ходе моделирования имеется возможность наблюдать за процессами, происходящими в системе. Для этого используются специальные смотровые окна, входящие в библиотеку SimuLink.

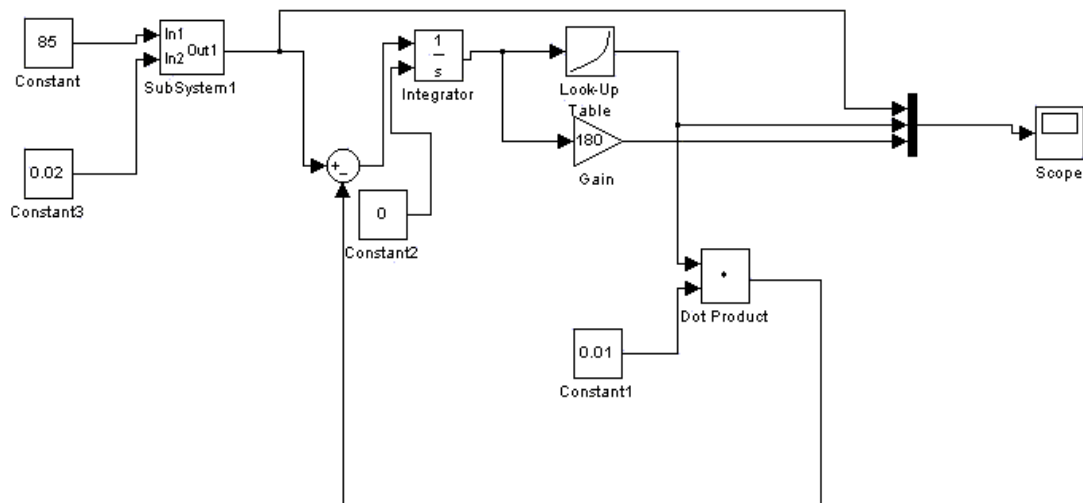


Рисунок 12.1 – S- модель (основная модель) задачи

S-модель для рассматриваемой задачи приведена на рис. 12.1. Она состоит из следующих блоков:

1. Субмодель для расчета напряжения „SubSystem1”;
2. Блок „Integrator” для решения дифференциального уравнения (12.1);
3. Блок для определения $i(\Psi)$ с помощью кусочно-линейной интерполяции „Look-up Table”;
4. Блок „Scope” для просмотра зависимостей $i(t)$, $u(t)$, $\Psi(t)$;
5. Блок „Dot Product” для умножения $i \cdot R$;
6. Блок „Sum” для вычисления $U - i \cdot R$; Полученный сигнал поступает на вход блока „Integrator”;
7. Блок „Max” создает вектор из трех сигналов, который поступает в блок „Scope”;
8. Блоки „Constant” являются исходными данными для расчета. Заданы амплитуда напряжения, период, сопротивление катушки и начальное потокосцепление. Данные для магнитной характеристики являются параметрами блока „Look-up Table”.

Субмодель для расчета напряжения приведена на рис. 12.2. Она реализует следующий алгоритм: если $t \geq k \cdot T$ и $t \leq k \cdot T + T/2$, то $U = U_Z$, иначе

$U = -UZ$. Здесь T – период, t – текущее время (определяется с помощью блока „Clock”), $k = \text{целое}(t/T)$.

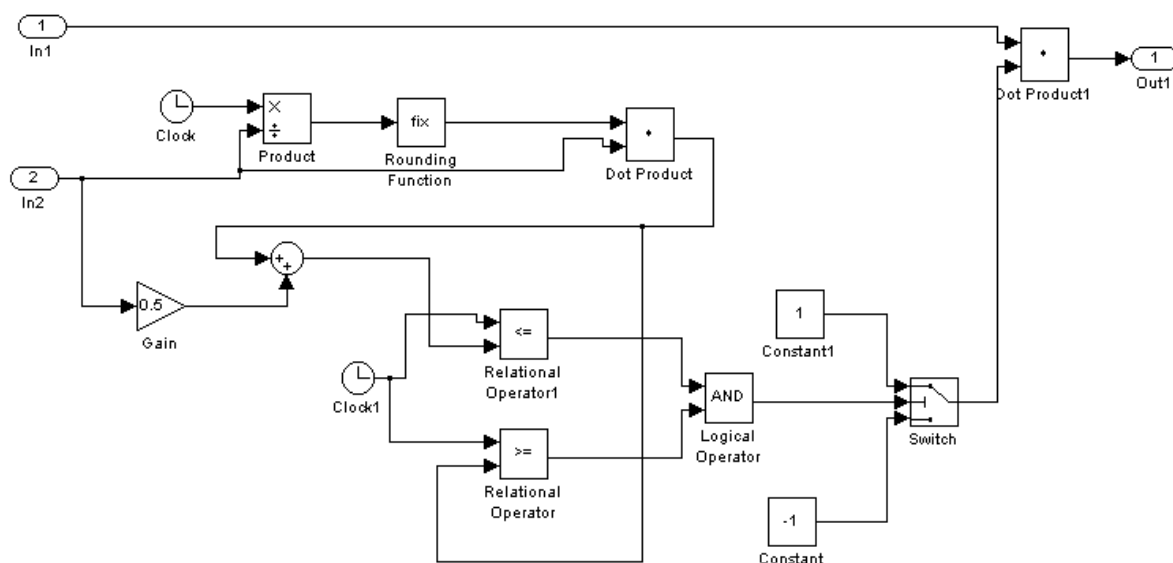


Рисунок 12.2 – Субмодель для определения напряжения

На вход субмодели поступают сигналы UZ и T . С помощью блоков „Product” (деление t/T), „Rounding Function” (округление до целого) и „Dot Product” (умножение на T) определяем $k \cdot T$, затем суммируем этот сигнал с сигналом $T \cdot 0,5$ (блоки „Sum” и „Gain”). Полученные сигналы поступают на логические блоки „Relation Operation”, где проверяются условия $t \geq k \cdot T$ и $t \leq k \cdot T + T/2$. Выходы этих блоков подсоединяются к блоку „Logical Operator”, который выполняет в данном случае операцию логического сложения AND. Далее сигнал поступает на средний вход переключателя „Switch” и сравнивается с пороговым сигналом блока (задано 0,5). Если оба условия выполняются, то поступающий сигнал, равный 1, будет больше порогового, и на выход блока будет подан сигнал с верхнего входа блока, т.е. 1. Иначе поступающий сигнал, равный 0, будет меньше порогового, и на выход блока будет подан сигнал с нижнего входа блока, т.е. -1 . Сигнал с выхода блока „Switch” умножается на напряжение (блок „Dot Product”) и поступает на выход субмодели Out1.

12.4. Содержание лабораторной работы

Цель работы – сравнить два способа решения дифференциальных уравнений с помощью программы на языке Паскаль и с помощью модели в программе SimuLink; сделать выводы о достоинствах и недостатках каждого из них.

Программа работы:

1. Провести расчеты по программе, описанной в подразделе 12.2, на следующие исходные данные: $U_z=85$; $T=0,02$; $R=0,01$; $X_k=0,04$; $h=0,001$.

2. Зафиксировать результаты расчета в конечной точке i_k, ψ_k и занести в табл. 12.1.

3. Повторить расчеты для шагов $h=0,0005$ и $h=0,00025$. Затем повторить расчеты для метода Рунге-Кутты 4 порядка и также занести в табл. 12.1.

Таблица 12.1 – Результаты расчета

Метод	$h=0,001$		$h=0,0005$		$h=0,00025$	
	i_k	ψ_k	i_k	ψ_k	i_k	ψ_k
Эйлера						
RK-4						

4. Зайти в программу SimuLink (пакет програм Matlab). Собрать основную модель и субмодель в соответствии с рис.12.1 и.12.2 и запустить процесс с помощью кнопки Simulation, предварительно установив время процесса. Открыть смотровое окно Scope и наблюдать процесс, определить приблизительно i_k, ψ_k . Для точного определения этих параметров установить в модели два блока „Display”(раздел Sink). Сравнить результаты расчета на модели с данными табл. 12.1. Сделать выводы.

5. Изменить в модели значения напряжения и сопротивления, наблюдать процессы и сделать вывод о влиянии этих параметров на характер процесса, в частности о влиянии насыщения на характер тока.

Контрольные вопросы

1. Запишите функцию для кусочно-линейной интерполяции зависимости $i(\Psi)$.
2. Запишите функцию для расчета правой части дифференциального уравнения (12.1) в форме Коши.
3. Опишите действие блока Switch в модели для программы SimuLink.
4. Как задается время в модели для программы SimuLink?
5. Как называется блок для отображения характера изменения переменных в переходном процессе?
6. Как влияет насыщение на характер тока в катушке?
7. Как влияет величина сопротивления катушки на длительность переходного процесса?

ЛЕКЦИЯ 13

РАСЧЕТ ТОКА В ВЕНТИЛЬНО-ИНДУКТОРНОМ ДВИГАТЕЛЕ (ВИД)

(второй пример применения численных методов решения ОДУ в
электротехнических расчетах)

Описание конструкции и принципа действия ВИД.

Описание задачи.

Описание программы.

Содержание лабораторной работы.

13.1. Описание конструкции и принципа действия ВИД

ВИД имеет Z_c полюсных выступов на статоре и Z_R полюсных выступов на роторе. Широко применяется конструкция с $Z_c=8$ и $Z_R=6$. На статорных полюсах расположены сосредоточенные катушки. Катушки противоположных полюсов соединяются последовательно и образуют фазу. Число фаз $m=Z_c/2$.

Катушки фаз поочередно подключаются к источнику постоянного напряжения в момент времени, когда очередной полюс ротора подходит к полюсу статора данной фазы и угол между полюсами принимает определенное значение в пределах $\Theta_{\text{вкл}}=30 - 22^\circ$. Предыдущая фаза может быть отключена в момент включения следующей или несколько позже. Как правило, угол отключения $\Theta_{\text{откл}}>0$.

При подключении фазы к источнику протекает ток, создается поток, намагничиваются полюса статора и ротора, и между ними возникает сила притяжения, под действием которой ротор будет вращаться.

При вращении ротора уменьшается угол между полюсами, увеличиваются магнитная проводимость, поток и потокоцепление катушек фазы. Таким образом, в ВИД потокоцепление является функцией не только тока, но и угла между полюсами, т.е. $\Psi=f(I, \Theta)$.

13.2. Описание задачи

Для определения тока фазы необходимо решить дифференциальное уравнение

$$U = R \cdot i(\Psi, \Theta) + \frac{d\Psi}{dt} \quad (13.1)$$

с начальным условием

$$t=0; \psi=0. \quad (13.2)$$

Для решения уравнения (13.1) необходимо иметь магнитные характеристики, т.е. зависимости $\Psi=f(I, \Theta)$. Их получают путем расчета магнитного поля методом конечных элементов с помощью готовых пакетов программ, например пакета *FEMM*. Магнитные характеристики (другое название кривые намагничивания) для углов 0; 6,5; 13; 19,5; 26; 30 град приведены на рис. 13.1.

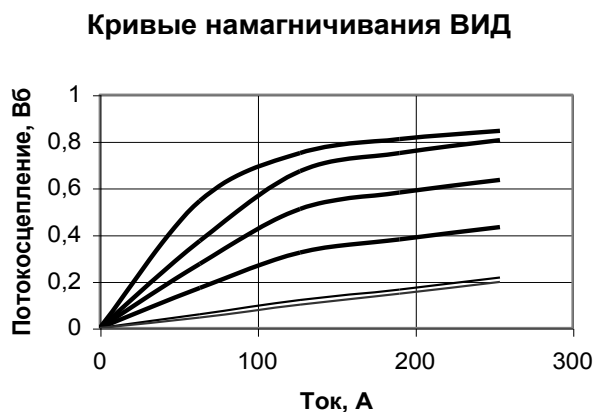


Рисунок 13.1 – Магнитные характеристики

Информация о магнитных характеристиках вводится в программу в виде массивов: двумерного массива $\text{PSI}[0..5, 0..4]$ и одномерных массивов $\text{TETA}[0..5]$ и $\text{ITI}[0..4]$. Значение тока для данных значений потокосцепления и угла определяем с помощью функции двумерного интерполирования **Function RP**, в которой используется процедура одномерного кусочно-линейного интерполирования **Procedure KUSLIN**.

Алгоритм двумерного интерполирования имеет следующий вид:

1. Для всех табличных значений тока с помощью процедуры кусочно-линейного интерполирования определяем значения потокосцеплений Ψ для заданного угла Θ_3 . Таким образом, получаем дополнительную магнитную характеристику $\Psi(\Theta_3, i)$;

2. Используя эту магнитную характеристику, с помощью процедуры кусочно-линейного интерполирования для заданного потокосцепления находим ток.

Расчет тока проводим для установившегося режима, т.е. при $\Omega = \text{const}$, где Ω – угловая скорость двигателя. В этом случае удобно задаваться шагом интегрирования в виде приращения угла $d\theta$, выраженного в градусах, а затем в процедуре Рунге-Кутты пересчитать шаг с помощью соотношения

$$H_t = h_\theta / \Omega / 180 \cdot 3,14. \quad (13.3)$$

Для решения уравнения (13.1) используем метод Рунге-Кутты 4 порядка. Уравнение представим в форме Коши

$$\frac{d\Psi}{dt} = U - R \cdot i(\Psi, \Theta). \quad (13.4)$$

Здесь $U > 0$ при $\Theta > \Theta_{\text{откл}}$ и $U < 0$ при $\Theta < \Theta_{\text{откл}}$, так как после размыкания электронных ключей, подающих питание на катушки фаз, ток возвращается в источник через обратные диоды навстречу напряжению.

13.3. Описание программы

Составим *программу для расчета тока* и построения графиков зависимостей тока, потокосцепления и напряжения от времени на периоде.

Программа состоит из следующих структурных элементов:

1. Функции **Function** RP для двумерного интерполирования, которая включает в себя процедуру **Procedure** KUSLIN для одномерного интерполирования;

2. Процедуры **Procedure** RPK4 для вычисления правой части уравнения (13.4);

3. Процедуры **Procedure** RK4 для решения дифференциального уравнения (13.4);

4. Процедуры **Procedure GRAFIKA** для построения графиков функций $i(t)$, $\Psi(t)$, $u(t)$.

Исходные данные для расчета:

1. Таблица для магнитных характеристик;
2. Амплитуда питающего напряжения UZ ;
3. Активное сопротивление катушки R ;
4. Углы $\Theta_{\text{вкл}}$ и $\Theta_{\text{откл}}$;
5. Угловая скорость Ω ;
6. Шаг интегрирования по углу в градусах h .

Далее приводим текст программы с комментариями.

Program VID;

Uses

Crt, Graph;

Type MATR=Array[0..5,0..4] Of Real;

Const

PSI:MATR=((0, 0.3767,0.50937, 0.5548, 0.59), (0,0.30075,
0.45895, 0.5085,0.5365), (0, 0.20335, 0.3277, 0.373, 0.40885), (0, 0.103, 0.175,
0.2183, 0.2581), (0, 0.042, 0.08, 0.116, 0.152), (0, 0.031625, 0.06325, 0.0949,
0.12655));

ITI:Array[0..4] **Of Real**= (0, 80, 160, 240, 320);

TETA:Array[0..5] **Of Real**=(0, 6.5, 13, 19.5, 26, 30);

Var

it, y, xt, u, r, h, xot, fx, xt1, uz, itp, omega, itef:**Real**;

i, j, np, n: **Integer**;

IT1, PSI1, UP:Array[0..1000] **Of Real**;

Function RP(y:Real; xt1:Real):Real;

Var

F2:Array[0..5] **Of Real**;

F1:Array[0..4] **Of Real**;

i, j: **Integer**;

**Procedure KUSLIN(f, x: Array Of Real; n:Integer;x1:Real; Var
fx:Real);**

```

Var
  i: Integer;
Begin
  i:=0;
  Repeat i:=i+1 Until x1<=x[I];
  fx:=f[i-1]+(x1-x[i-1])*(f[i]-f[i-1])/(x[i]-x[i-1]);
End;
Begin {головной модуль функции RP}
  For i:=1 To 4 Do Begin
    For j:=0 To 5 Do
      F2[j]:=PSI[j,i];
      KUSLIN(f2,teta,5,xt1,fx); F1[i]:=fx;
    End;
    F1[0]:=0;
    KUSLIN(ITI, F1,4,y,it);
    RP:=it;
  End;
Procedure RPK4(xt:Real; Var itp, y, f,xt1:Real);
Begin
  xt1:=ABS(xt);
  If (xt<xot) Or (xt=xot) Then u:=-uz Else u:=uz;
  itp:=RP(y, xt1);
  f:=u-itp*r;
End;
Procedure RK4(Var xt,h,xt1,y:Real;xot:Real);
Var
  i,j:Integer;
  h1,h2,q:Real; y0,y1,f,kth:Real;
Begin
  kth:=3.14/omega/180;{коэффициент для определения
  h1:=0; шага по времени}
  h2:=h/2;y0:=y;y1:=y;

```

```

For j:=1 To 4 Do
Begin
  RPK4(xt-h1, itp, y, f, xt1);
  If j=3 Then h1:=h Else h1:=h2;
  q:=h1*f*kth; y:=y0+q;
  If j=2 Then q:=2*q; y1:=y1+q/3.0;
  End;
  y:=y1;
End;
Procedure GRAFIKA(n:Integer);
Var
  i, Gd, Gm:Integer;
  mi, mpsi, mu, mx:Real;
Begin
  Gd:=Detect;
  Initgraph(Gd, Gm, ' ');
  mi:=240/320; mpsi:=240/0.9; mx:=640/n; mu:=180/uz;
  For i:=1 To n Do Putpixel(Round(i*mx), 240, 1);
  For i:=1 To j Do Putpixel(Round(i*mx), 240-
    Round(IT1[i]*mi), 3); {построение графика для тока}
  For i:=1 To j Do Putpixel(Round(i*mx), 240-
    Round(PSI1[i]*mpsi), 10); {построение графика для потокос-
цепления}
  For i:=1 To j Do Putpixel(Round(i*mx), 240-
    Round(UP[I]*mu), 12); {построение графика для напряжения}
  End;
Begin
  Writeln('xt xot h omega u r');
  Readln(xt, xot, h, omega, uz, r);
  j:=0;
  PSI1[0]:=0; IT1[0]:=0; UP[0]:=uz; itef:=0;
  Repeat

```

```

j:=j+1; PK4(xt,h,xt1,y,xot);{вызов процедуры PK4}
PSI1[j]:=y; IT1[j]:=RP(y,xt1);
UP[J]:=u; itef:=itef+IT1[j]*IT1[j];
Writeln('xt',xt:5:2,' y',y:10:5,' itp',itp:10:5);
If Round(j/20)=j/20 Then Readln;
xt:=xt-h;
Until y<0;
np:=Round(60/h);
itef:=Sqrt(itef/np);{эффективное значение тока}
Writeln('itef=',itef:10:5);
Readln;
GRAFIKA(np);
Repeat Until Keypressed;
Closegraph;
End.

```

13.4. Содержание лабораторной работы

Цель работы – исследовать влияние амплитуды питающего напряжения U_z , углов включения и отключения $\Theta_{\text{вкл}}$ и $\Theta_{\text{откл}}$ на величину и форму тока при данной угловой скорости Ω .

Программа работы:

1. Задаемся значениями $U_z = 80$ В; $R = 0,05$ Ом; $\Omega = 65$ рад/с; $h = 0,2$ град. Для значений $\Theta_{\text{вкл}}$ и $\Theta_{\text{откл}}$, указанных в табл. 13.1, проводим расчеты, фиксируем эффективное значение тока $I_{\text{эф}}$ и заносим в табл. 13.1. Полученные графические рисунки запоминаем в буфер, восстанавливаем в программе Paint и записываем в файл, затем печатаем на принтере и вставляем в отчет по лабораторной работе.

Таблица 13.1 – Результаты расчета

$\Theta_{\text{вкл}} - \Theta_{\text{откл}}$	30 – 11	28 – 9	26 – 7
$I_{\text{эф}}$			

2. Повторяем расчеты для напряжений $U_z = 60 \text{ В}$ и $U_z = 100 \text{ В}$;
3. Делаем выводы о влиянии U_z , $\Theta_{\text{вкл}}$ и $\Theta_{\text{откл}}$ на величину эффективно-го тока и на форму тока.

Контрольные вопросы

1. Опишите конструктивное исполнение вентильно-индукторного двигателя (ВИД).
2. Опишите принцип действия ВИД.
3. Опишите алгоритм двумерной интерполяции.
4. Опишите процедуру для расчета правой части дифференциального уравнения (13.1) в форме Коши.
5. Как рассчитать эффективное значение тока?
6. Как влияет изменение значения $\Theta_{\text{вкл}}$ при постоянной разности $\Theta_{\text{вкл}} - \Theta_{\text{откл}}$ и при постоянном напряжении на величину эффективного значения тока и на форму тока?
7. Как влияет изменение величины напряжения при постоянных значениях $\Theta_{\text{вкл}}$ и $\Theta_{\text{откл}}$ на величину эффективного тока и на его форму?

ЛЕКЦИЯ 14

МЕТОДЫ ПОИСКА ЭКСТРЕМУМОВ ФУНКЦИЙ ОДНОЙ ПЕРЕМЕННОЙ

Общие положения.

Метод поразрядного приближения.

Метод дихотомии.

Метод золотого сечения.

Описание задачи поиска минимума стоимости активной части масляного трансформатора.

Содержание лабораторной работы.

14.1. Общие положения

При проектировании электрических машин обязательно проводится оптимизация по одному какому-либо критерию, который называется параметром оптимизации (это или минимум стоимости, или максимум КПД, или минимум приведенных затрат). Оптимальное решение является функцией множества переменных. Однако при поиске оптимума функции множества переменных часто используют методы поиска оптимума функции одной переменной.

На заданном интервале изменения переменной может находиться несколько точек оптимума, один из которых является глобальным. Задача поиска оптимума разбивается на два этапа. На первом этапе выделяются участки, на которых имеется один оптимум. Функция на этом участке называется „унимодальной”. На втором этапе уточняется значение оптимума. Далее рассмотрим методы для второго этапа.

Предполагаем, что необходимо найти максимум функции. Так как максимум функции $F(x)$ соответствует минимуму функции $-F(x)$, то программа для поиска максимума может быть использована для поиска минимума, если заменить знак у функции противоположным.

Простейшим методом поиска оптимума является **метод равномерного поиска**, в котором используется постоянный шаг для изменения значения аргумента. На каждом шаге дается приращение аргументу функции $x + \Delta x$ и

производится сравнение двух значений функции. Если $F(x_n) > F(x_{n-1})$, где n – номер шага, то поиск оптимума продолжается. Если $F(x_n) < F(x_{n-1})$, то поиск прекращается и оптимальным считается значение функции $F(x_{n-1})$. Для получения высокой точности определения оптимума необходим очень маленький шаг. Метод применяется для грубой оценки значения оптимума.

Далее рассмотрим три метода, которые позволяют существенно сократить время поиска оптимума и одновременно увеличить точность.

14.2. Метод поразрядного приближения

Алгоритм метода поразрядного приближения:

1. Задаемся начальным значением аргумента $x = x_0$;
2. Задаемся начальным значением шага $h = \Delta x = d$;
3. Вычисляем значение функции в точке $x = x_0$ и присваиваем это значение какой-либо переменной $G = F(x_0)$ (на следующих шагах $G = F(x)$);
4. Делаем приращение аргумента $x + d$, вычисляем $F(x + d)$ и сравниваем $F(x + d) > G$. Если „да”, идем к п. 3, иначе – к п. 5;
5. Вычисляем новое значение шага $h = -d/4$;
6. Если $d/4 > e_{\text{доп}}$, то идем к п. 3, иначе – к п. 7 (здесь $e_{\text{доп}}$ – допустимое значение погрешности при определении оптимума);
7. Печать координат точки оптимума x_m , $F(x_m)$.

В соответствии с этим алгоритмом составляем процедуру.

Procedure PORAZR;

Var

edop, p, d, x, g, h: **Real**;

Begin

Writeln('x h edop '); {ввод начального значения x, шага h

Readln(x, h, edop); и допустимой погрешности}

n:=0; {обнуление счетчика числа обращений к функции}

d:=h;

Repeat

Repeat

g:=f(x); x:=x+d;

```

Until f(x)<g;
    d:=-d/4;
Until ABS(d)<4*edop;
Writeln ('x=',x:10:5, ' f(x)=',f(x):10:5, ' n=',n:5);
End;

```

14.3. Метод дихотомии (деления отрезка пополам)

Алгоритм метода дихотомии:

1. Интервал, в котором необходимо найти оптимум, делится пополам и определяются точки слева и справа от середины отрезка по формулам

$$X_1=(a + b - e_{\text{доп}})/2; X_2=(a + b + e_{\text{доп}})/2;$$

2. Вычисляются значения функций в этих точках и сравниваются между собой;

3. Если $F(X_1) < F(X_2)$, то принимаем $a = X_1$, иначе полагаем $b = X_2$. Таким образом, уменьшаем размеры интервала;

4. Пункты 1, 2, 3 повторяем до тех пор, пока не будет выполнено условие $|b - a| < 2 \cdot e_{\text{доп}}$;

5. Оптимальное значение аргумента рассчитываем по формуле

$$X_m=(a + b) / 2.$$

В соответствии с этим алгоритмом составляем процедуру.

Procedure DIHOTO;

Var

a,b,x1,x2,xm,edop:**Real**;

Begin

Writeln('a b edop');

Readln(a,b,edop);

n:=0;

Repeat

x1:=(a+b-edop)/2; x2:=(a+b+edop)/2;

If f(x1)<f(x2) **Then** a:=x1 **Else** b:=x2;

Until ABS(b-a)<2*edop;

xm:=(a+b)/2;

```

Writeln('xm=',xm:10:5,' f(xm)=',f(xm):10:5,' n=',n:5);
End;

```

14.4. Метод золотого сечения

В этом методе по сравнению с методом дихотомии число обращений к функции уменьшается в два раза за счет разбиения интервала на три части в специальной пропорции.

Алгоритм метода золотого сечения:

1. Вычисляем координаты точек X_1, X_2 по формулам

$$X_1 = a + (1 - k)(b - a);$$

$$X_2 = a + k(b - a),$$

где $k = 0,618034$.

2. Вычисляем значения функций в этих точках и сравниваем их между собой;

3. Если $F(X_1) < F(X_2)$, принимаем $a = X_1$; $X_1 = X_2$; $F(X_1) = F(X_2)$ и вычисляем новое значение X_2 по формуле п.1 и $F(X_2)$, иначе принимаем $b = X_2$; $X_2 = X_1$; $F(X_2) = F(X_1)$ и вычисляем новое значение X_1 по формуле п.1 и $F(X_1)$;

4. Повторяем п.3 до тех пор, пока не будет выполнено условие

$$|b - a| < e_{\text{доп}} \cdot 2;$$

5. Вычисляем оптимальное значение аргумента по формуле $X_m = (X_1 + X_2)/2$.

В соответствии с описанным алгоритмом составляем процедуру ZOLOTO.

Procedure ZOLOTO;

Var

a,b,x1,x2,xm,f1,f2edop:**Real**;

Const

k=0.618034;

Procedure FUN1; ;{описание процедуры FUN1 для расчета $X1$

Begin и F1 }

```

        x1:=a+(1-k)*(b-a);
        f1:=f(x1);
    End;
Procedure FUN2;{описание процедуры FUN2 для расчета x2
    Begin                и f2 }
        f2:=f(x2);
        x2:=a+x*(b-a);
    End;
Begin
    Writeln('a b edop'); Readln(a,b,edop);
    n:=0;
    FUN1; {вызов процедуры FUN1}
    FUN2; {вызов процедуры FUN2}
    Repeat
        If f1<f2 Then
            Begin
                a:=x1; x1:=x2; f1:=f2; FUN2; ;{вызов процедуры
FUN2}
            End
        Else
            Begin
                b:=x2; x2:=x1; f2:=F1; FUN1; {вызов процедуры
FUN1}
            End;
        Until ABS(b-a)<edop;
        xm:=(x1+x2)/2;
        Writeln('xm=',xm:10:5,' f(xm)',f(xm):10:5,' n',n:5);
    End;

```

14.5. Описание задачи поиска минимума стоимости активной части масляного трансформатора

Стоимость активной части масляного трансформатора существенным образом зависит от отношения средней длины витка двух обмоток к высоте обмотки. Это отношение обозначается коэффициентом β . Введем параметр $x = \sqrt[4]{\beta}$. Тогда формула для зависимости $C_{\text{акт}} = f(x)$ будет иметь вид

$$C_{\text{акт}} = A1 \cdot x^3 + B1 \cdot x^2 + C1 / x + D1 / x^2 \quad (14.1)$$

Здесь $A1, B1, C1, D1$ – коэффициенты. Для отладки программы предлагаем два варианта значений коэффициентов из учебника по проектированию трансформаторов П.М.Тихомирова для трансформатора мощностью 1600 кВ·А с медными и алюминиевыми обмотками: 1) $A1 = 529,7$; $B1 = 183,5$; $C1 = 800,9$; $D1 = 1807$; 2) $A1 = 648$; $B1 = 205,8$; $C1 = 960,26$; $D1 = 1029$. Точное решение уравнения (14.1) дает $\beta_{\text{опт}} = 2,14$ для первого варианта и $\beta_{\text{опт}} = 1,36$ для второго варианта. В уравнении (14.1) первые три слагаемые определяют стоимость магнитопровода, а последнее слагаемое – стоимость обмоток. Изменяя соответствующим образом значения коэффициентов можно определить, как влияет отношение стоимости обмоток к стоимости магнитопровода на величину $\beta_{\text{опт}}$.

Для расчета $C_{\text{акт}}$ составим функцию, которую приводим далее.

Function F(x:Real):Real;

Var

z:Real;

Begin

$n:=n+1$; {считаем число обращений к функции}

$z:=\text{Exp}(0.25 \cdot \text{Ln}(x))$;

$f:=-(A1 \cdot z \cdot \text{SQR}(z) + B1 \cdot \text{SQR}(z) + C1/z + D1/\text{SQR}(z))$; {ставим "–",

так как необходимо найти минимум функции, а программа составлена для поиска максимума}

End;

14.6. Содержание лабораторной работы

1. Составляем программу для сравнительной оценки методов поиска оптимума. Программа должна содержать описание счетчика числа обращений к функции, самой функции и трех процедур, а в исполнительной части – обращение к этим процедурам.

2. С помощью составленной программы производим поиск минимума для одного варианта данных в диапазоне изменения $\beta=1 - 3,6$ (в программе $a = 1, b = 3,6$) с точностью $e_{\text{доп}} = 0,01$. Начальное значение β (x в программе) и начальное значение шага h в методе поразрядного приближения выбираем произвольно. Результаты заносим в табл. 14.1.

Таблица 14.1 – Результаты расчета

Метод	Поразрядного приближения	Дихотомии	Золотого сечения
Число обращений к функции			

По данным табл. 14.1 выбираем наилучший метод по критерию наименьшего числа обращений к функции.

4. Уменьшаем и увеличиваем значение коэффициента $D1$ по сравнению с исходным на 20 %, производим расчет по наилучшему методу и делаем вывод, как влияет уменьшение и увеличение стоимости обмоточного провода на величину $\beta_{\text{опт}}$.
5. Разрабатываем процедуры для графической иллюстрации методов оптимизации, с помощью которых можно наблюдать движение к оптимуму в методе поразрядного приближения и уменьшение интервала для поиска оптимума в методах дихотомии и золотого сечения.

Контрольные вопросы

1. Опишите алгоритм метода поразрядного приближения.
2. Опишите алгоритм метода дихотомии.

3. Опишите алгоритм метода золотого сечения.
4. Опишите задачу поиска минимума зависимости стоимости активной части трансформатора от коэффициента β .
5. В чем преимущество метода золотого сечения по сравнению с методом дихотомии.
6. Замените в программах циклы с послеусловием на циклы с предусловием.
7. Опишите алгоритм метода равномерного поиска и назовите его достоинства и недостатки.
8. Составьте процедуру для метода равномерного поиска.
9. Как влияет изменение стоимости обмоточного провода на величину $\beta_{\text{опт}}$?

ЛЕКЦИЯ 15

ПОИСК ОПТИМУМА ФУНКЦИЙ НЕСКОЛЬКИХ ПЕРЕМЕННЫХ

Метод координатного спуска (описание метода).

Процедура для метода координатного спуска.

Программа для поиска минимума зависимости приведенных годовых затрат трансформатора от индукции и плотности тока.

Метод крутого восхождения.

Содержание лабораторной работы.

15.1. Метод координатного спуска (описание метода)

Рассмотрим **алгоритм поиска минимума** многомерной функции на примере функции двух переменных $f(Z1, Z2)$. График функции $f(Z1, Z2)$ в области её минимума представим в параметрическом виде, подобно изображению рельефа местности на географических картах, соединяя линиями точки на координатной плоскости $(Z1, Z2)$, где функция принимает одинаковые значения.

Предположим, что нам известна прямоугольная область на плоскости $(Z1, Z2)$, где находится минимум функции $f(Z1, Z2)$, т.е.

$$Z1_{\text{опт}} \in [Z1_n, Z1_k], \quad Z2_{\text{опт}} \in [Z2_n, Z2_k].$$

Алгоритм координатного спуска заключается в сведении многомерной задачи к последовательным одномерным задачам, которые решаются методами минимизации функции одной переменной. Если в области минимума функция достаточно гладкая, то процесс спуска по координатам будет линейно сходиться к минимуму. В сходящемся процессе с приближением к минимуму функции расстояние между последовательными точками однокоординатных минимумов будут стремиться к нулю.

Рассмотрим процесс движения к оптимуму с помощью рис. 15.1. Задаемся начальным значением Z_2^0 и фиксируем его. Тогда двумерная функция становится одномерной, т. е. $F(Z_1, Z_2^0) = f(Z_1)$. С помощью метода "золотого сечения" в заданном диапазоне изменения $Z_1 - (Z_{1n} - Z_{1k})$ при фиксированном значении $Z_2 = Z_2^0$ находим Z_1^0 – значение Z_1 , соответствующее ми-

нимуму функции. Фиксируем Z_1^0 . С помощью метода "золотого сечения" в заданном диапазоне изменения $Z_2 - (Z_{2n} - Z_{2k})$ при фиксированном значении $Z_1 = Z_1^0$ находим Z_2^1 , соответствующее минимуму функции, и так далее, пока не будет выполнено условие

$$|Z_1^k - Z_1^{k-1}| < \varepsilon_{\text{доп}}; \quad |Z_2^k - Z_2^{k-1}| < \varepsilon_{\text{доп}},$$

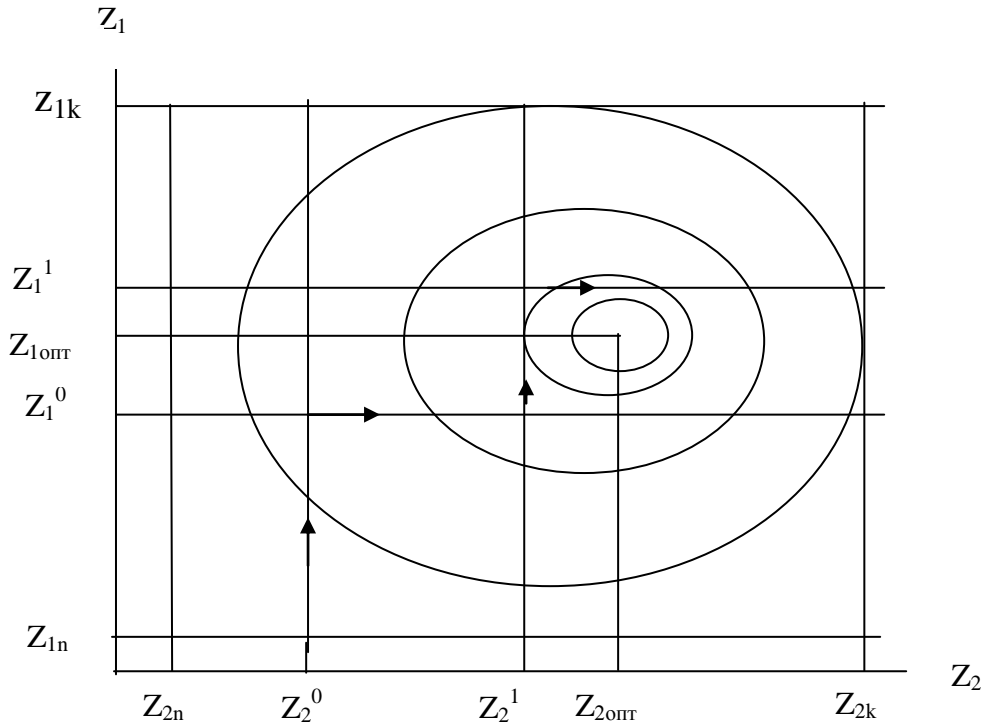


Рис. 15.1 – Иллюстрация к методу координатного спуска

где $\varepsilon_{\text{доп}}$ – допустимая погрешность определения точки минимума. При большом числе переменных (больше двух) метод координатного спуска требует больших затрат времени на поиск точки оптимума. Более быстрым является градиентный метод и его модификация – „метод крутого восхождения”, который будет рассмотрен в подразделе 15.4.

15.2. Процедура для метода координатного спуска

В соответствии с рассмотренным алгоритмом составим процедуру для метода координатного спуска. Далее приводим процедуру с комментариями.

Procedure Coord;

Begin

For i:=2 **to** n **do** z[i]:=z1[i]; *{присваиваем переменным, начиная со второй, их начальные значения}*

l:=0;

Repeat

For i:=1 **to** n **do Begin**

a:=zn[i]; b:=zk[i]; *{определяем границы интервала для поиска минимума методом золотого сечения}*

zoloto; *{вызываем процедуру метода золотого сечения}*

If ABS(x- z1[i]) > edop **Then** l:=1; z1[i]:=x; **End;**

Until l=0;

End;

15.3. Программа для поиска минимума зависимости приведенных годовых затрат трансформатора от индукции и плотности тока

При проектировании электрических машин, в том числе и трансформаторов, основным критерием при поиске оптимального решения является минимум приведенных годовых затрат – $Z_g = \min$. Они учитывают затраты на производство изделия и затраты на его эксплуатацию, которые зависят от потерь в изделии. При увеличении индукции в стержне B_c и плотности тока в обмотках J , с одной стороны, растут потери и затраты на эксплуатацию, с другой стороны, уменьшаются габариты изделия и затраты на его изготовление. Поэтому зависимость $Z_g = f(B_c, J)$ имеет минимум при определенном соотношении параметров $B_{c\text{опт}}$ и $J_{\text{опт}}$.

Для упрощения задачи воспользуемся математической моделью для этой зависимости для трансформатора мощностью 2500 кВ·А, разработан-

ной студентами в процессе игрового проектирования с помощью метода математического планирования эксперимента. Модель имеет вид

$$Z_r = 1889 - 7,66 \cdot x_1 + 15,66 \cdot x_2 + 13,85 \cdot x_1^2 + 69,29 \cdot x_2^2 - 4,5 \cdot x_1 \cdot x_2. \quad (15.1)$$

Здесь x_1, x_2 – кодированные факторы. Модель разработана для следующих диапазонов изменения реальных факторов:

$$X_1(B_c) = 1,5 - 1,7 \text{ Тл}, \quad X_2(J) = 1 - 2 \text{ А/мм}^2.$$

После определения оптимальных значений $x_{1\text{опт}}, x_{2\text{опт}}$ для кодированных факторов рассчитываем оптимальные значения $X_{1\text{опт}}, X_{2\text{опт}}$ для реальных факторов по соотношениям

$$X_{1\text{опт}} = x_{1\text{опт}} \cdot I_1 + X_{1\text{ср}}; \quad X_{2\text{опт}} = x_{2\text{опт}} \cdot I_2 + X_{2\text{ср}}.$$

Здесь $I_1=0,1$ – интервал для первого фактора; $I_2 = 0,5$ – интервал для второго фактора; $X_{1\text{ср}}=1,6$ – среднее значение первого фактора; $X_{2\text{ср}} = 1,5$ – среднее значение для второго фактора.

Далее приводим программу с комментариями. Программа включает следующие структурные элементы:

1. Функцию Function F для определения приведенных затрат по уравнению (15.1);
2. Процедуру Zoloto для поиска минимума функции одной переменной;
3. Процедуру Coord для поиска минимума функции нескольких переменных.
4. Процедуру Grafika для иллюстрации перемещения расчетных точек при движении к оптимуму.

Program COORDSPU;

Uses Crt, Graph;

Type vec=Array[1..8] **of** Real;

Const k=0.618034;{коэффициент для метода золотого сечения}

Var

i,n,m,nz,l:**Integer**;

x,a,b,ez,ec,x1,x2,f1,f2,r,v:**Real**;

int,xcpr,zr:vec;{интервалы, средние и оптимальные значения реальных

факторов}
*z,zn,zk,z1:vec;{текущие, граничные и начальные значения перемен-
ных}*
*y1,w1,y,w:array[0..1000] of real;{массивы координат точек для гра-
фика}*
Function f(x:real):**Real**;{функция для расчета приведенных годовых за-
трат трансформатора}
Begin
 z[i]:=x; m:=m+1; y[m]:=z[1]; w[m]:=z[2];
 *r:=1889- 7.66*z[1]+15.66*z[2]+13.85*sqr(z[1])+69.29*sqr(z[2])–*
 *4.5*z[1]*z[2]; f:=r;*
End;
Procedure zoloto; *{процедура для метода золотого сечения- вставля-*
ем из предыдущей лекции}
Procedure coord;{процедура для метода координатного спуска}
{перепишем из предыдущего подраздела, добавляем расчет счетчиков
nz,m,
массивов y, w- координат точек при движении к оптимуму, массивов
y1, w1- координат точек минимумов в данном направлении}
Begin
 For i:=2 **to** n **do** z[i]:=z1[i];
 nz:=0;m:=0;{обнуляем счетчики nz – число изменений направления
 поиска оптимума; m – число обращений к функции}
 y[0]:=z1[1]; w[0]:=z1[2]; y1[0]:=z1[1]; w1[0]:=z1[2];
 Repeat
 l:=0;
 For i:=1 **to** n **do** **Begin**
 a:=zn[i]; b:=zk[i]; nz:=nz+1; zoloto;
 If abs(x-z1[i])>ec **Then** l:=1; z1[i]:=x;
 y1[nz]:=z1[1]; w1[nz]:=z1[2]; **End;**
 Until l=0;
 End;

Procedure grafika;

Var

Gd,Gm ,Errorcode:**Integer**; mx,my:**Real**;

Begin

Gd:=Detect;

InitGraph(Gd,Gm,' ');ClearDevice;

my:=480/(zk[2]-zn[2])/1.2; mx:=640/(zk[1]-zn[1])/1.2;

setbkcolor(15);

For i:=0 **To** 639 **Do** putpixel(i,240,14);

For i:=0 **To** 479 **Do** putpixel(320,i,14);

For i:=0 **To** m **Do Begin**

putpixel(320+round(mx*y[i]),240-round(my*w[i]),4); **End**;

For i:=0 **To** nz **Do Begin**

putpixel(320+round(mx*y1[i]),240-round(my*w1[i]),7); **End**;

End;

Begin {исполнительная часть головного модуля }

ClrScr;

Writeln('n ez ec');**Readln**(n,ez,ec);{ввод числа переменных, допустимых погрешностей для метода золотого сечения и метода координатного спуска}

For i:=1 **To** n **Do**

Begin

Writeln('i=',i:3,' zn zk z1');{ввод граничных и начальных значений

Readln(zn[i],zk[i],z1[i]); переменных}

End;

int[1]:=0.1;int[2]:=0.5;хср[1]:=1.6;хср[2]:=1.5;{задание интервалов и средних значений для переменных для модели (15.1)}

coord;{вызов процедуры для метода координатного спуска}

For i:=1 **To** n **Do Begin**

zr[i]:=z[i]*int[i]+хср[i]; {определение реальных факторов}

Writeln('i=',i:2,' z[i]=' z1[i]:5:2,' zr=',zr[i]:10:5);{печать оптимальных значений для кодированных и реальных факторов}

End;

Writeln('f=',r:7:3,' nz=',nz:2,' m=',m:4);{печать оптимального значения приведенных годовых затрат, числа изменений направления и числа обращений к функции}

Readln;

grafika;{вызов процедуры для графических построений}

Repeat Until Keypressed;

Closegraph;

End.

15.4. Метод крутого восхождения

Рассмотрим поиск максимума функции двух переменных. Для иллюстрации метода также как в п.15.1, изобразим линии равного уровня в заданной области (рис.15.2).

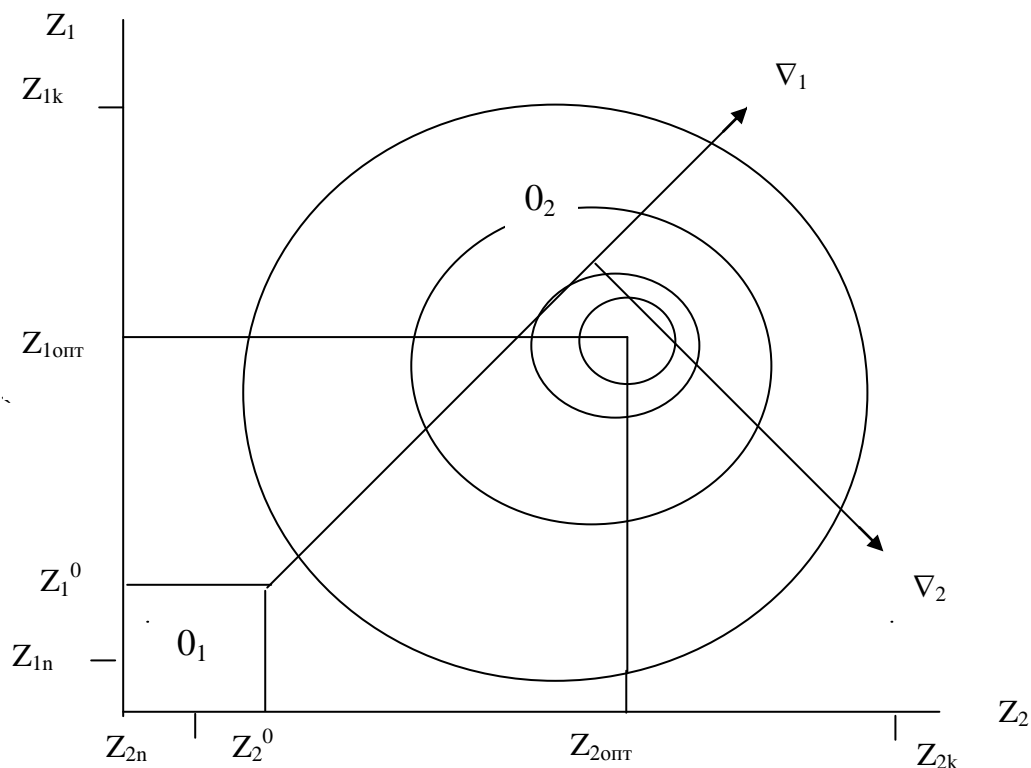


Рисунок 15.2 – Иллюстрация к методу крутого восхождения

Задаемся координатами начальной точки $O_1 - Z_1^0, Z_2^0$. В окрестности точки O_1 проводим полный факторный эксперимент ПФЭ², предварительно задаваясь интервалами I_1, I_2 . По результатам эксперимента строим линейную модель функции в окрестности точки O_1

$$F(x_1, x_2) = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2.$$

Определим градиент этой функции

$$\nabla F = i \cdot \frac{\partial F}{\partial x_1} + j \cdot \frac{\partial F}{\partial x_2} = i \cdot b_1 + j \cdot b_2.$$

Таким образом, составляющие градиента равны коэффициентам линейной модели. Используем эти коэффициенты для определения шагов при движении к оптимуму в направлении градиента:

$$h_{x1} = r \cdot b_1 \cdot I_1; h_{x2} = r \cdot b_2 \cdot I_2.$$

Здесь r – коэффициент пропорциональности. От его величины зависит скорость продвижения к оптимуму. При поиске минимума $r < 0$.

Изменяем исходные значения переменных на величину шагов:

$$Z_1^1 = Z_1^0 + h_{x1}; Z_2^1 = Z_2^0 + h_{x2}.$$

Находим значение функции F . Сравниваем новое и предыдущее значения функции. Если новое значение больше предыдущего, продолжаем движение в направлении градиента, прибавляя значения шагов к значениям переменных. Если новое значение станет меньше предыдущего, движение в направлении первого градиента прекращаем. Предыдущие значения переменных определяют координаты точки O_2 , в окрестности которой снова проводим полный факторный эксперимент и определяем новое направление градиента. Этот процесс движения в направлении градиента продолжаем до тех пор, пока разность соседних значений функции станет меньше допустимой.

Уточненное значение оптимума получаем из модели второго порядка, которую получаем с помощью *ОЦКП 2 порядка*.

15.5. Содержание лабораторной работы

1. По приведенной в п.15.3 программе производим расчет для поиска минимума зависимости приведенных годовых затрат трансформатора от индукции и плотности тока на следующие исходные данные:

$$zn[1] = -1; zk[1] = 1; zn[2] = -1; zk[2] = 1;$$

Для начальных значений $z1[1], z1[2]$ выбираем 3-4 варианта (например):

- 1) $z1[1] = z1[2] = -1$; 2) $z1[1] = z1[2] = 1$; 3) $z1[1] = z1[2] = 0$;
4) $z1[1] = 1; z1[2] = -1$; 5) $z1[1] = -1; z1[2] = 1$.

Также задаем несколько вариантов погрешностей:

- 1) $ez = ec = 0,01$; 2) $ez = ec = 0,005$; 3) $ez = ec = 0,001$.

Результаты расчета сводим в табл. 15.1.

Таблица 15.1 - Результаты расчета

№ варианта.	Исходные данные	m	Nz	$B_{\text{сопт}}$	$J_{\text{опт}}$	$3_{\text{гmin}}$
1	$z1[1] = z1[2] = ez = ec =$					
2						
3						

2. Нарисуйте в отчете график перемещения точек при движении к оптимуму для нескольких вариантов.

3. Составьте программу для метода крутого восхождения, используя материал лекций 4 и 5. Это задание может рассматриваться как зачетное и может быть выполнено дома.

Контрольные вопросы

1. Опишите алгоритм метода координатного спуска.
2. Нарисуйте графическую интерпретацию метода координатного спуска.
3. Опишите алгоритм метода крутого восхождения.

4. Нарисуйте графическую интерпретацию метода крутого восхождения.
5. Опишите структуру приведенных годовых затрат.
6. Поясните, почему зависимость $Z_r = f(B_c, J)$ имеет минимум.
7. Как определяется градиент в методе крутого восхождения?
8. С какой целью в методе крутого восхождения используются полный факторный эксперимент и ОЦКП 2 порядка?
9. Опишите структуру программы для поиска минимума приведенных годовых затрат трансформатора.

СПИСОК ЛИТЕРАТУРЫ

1. Мак-Кракен Д., Дорн У. Численные методы и программирование на Фортране. – М.: Изд-во „МИР”, 1977.
2. Фаронов В.В. ТурбоПаскаль. Начальный курс: Учеб. пособие. – М.: Нолидж, 2000.–.575 с.
3. Мудров А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран и Паскаль. – Томск: МП „РАСКО”, 1991.
4. Ивоботенко Б.А., Ильинский Н.Ф., Копылов И.П. Планирование эксперимента в электромеханике. – М.: Энергия, 1975.
5. Токарев Б.Ф. Электрические машины: Учеб. пособие для вузов. – М.: Энергоатомиздат, 1980. – 624 с.
6. Кузнецов В.А., Кузьмичев В.А. Вентильно-индукторные двигатели: Учеб. пособие. – М.: Изд-во МЭИ, 2003.
7. Галайко Л.П. Расчет тока и момента вентильно-индукторного двигателя различными методами // Вісник НТУ „ХПІ”. – Х. – 2001. – № 17.
8. Галайко Л.П. Имитационное моделирование установившихся режимов работы вентильно-индукторного двигателя // Електротехніка та електромеханіка. – 2005. –№ 1.
9. Гульятеев А.К. Имитационное моделирование в среде Windows. Практическое пособие. – СПб.: КОРОНА принт., 2001. – 400 с.
10. Дьяконов В.П. Справочник по алгоритмам и программам на языке Бейсик для персональных ЭВМ: Справочник. – М.: Наука. Гл. ред. физ.-мат. л-ры., 1989. – 240 с.
11. Тихомиров П.М. Расчет трансформаторов. – М.: Энергия, 1986.

СОДЕРЖАНИЕ

Предисловие.....	3
Лекция 1. Кусочно-линейная интерполяция функций одной переменной с постоянным шагом.....	5
Лекция 2. Кусочно-линейная интерполяция с переменным шагом.....	16
Лекция 3. Интерполяция сплайнами.....	23
Лекция 4. Интерполяция функций двух переменных с помощью математического планирования эксперимента (полный факторный эксперимент).....	31
Лекция 5. Интерполирование функций двух переменных с помощью математического планирования эксперимента (ортogonalный центрально-композиционный план 2 порядка).....	40
Лекция 6. Численное решение нелинейных уравнений.....	49
Лекция 7. Методы численного интегрирования и их применение для расчета магнитодвижущей силы зубца электрической машины.....	56
Лекция 8. Оценка погрешности интегрирования и интегрирование с заданной точностью.....	68
Лекция 9. Практический гармонический анализ.....	78
Лекция 10. Определение гармонического состава намагничивающего тока трансформатора.....	85
Лекция 11. Численное решение дифференциальных уравнений.....	96
Лекция 12. Расчет тока в катушке с железным сердечником при питании переменным напряжением прямоугольной формы.....	104
Лекция 13. Расчет тока в вентильно-индукторном двигателе.....	112
Лекция 14. Методы поиска экстремумов функций одной переменной.....	120
Лекция 15. Поиск оптимума функции нескольких переменных.....	128
Список литературы.....	138

Навчальне видання

ГАЛАЙКО Лідія Петрівна

**ВИКОРИСТАННЯ ЕОМ
В ЕЛЕКТРОТЕХНІЧНИХ
РОЗРАХУНКАХ**

Тексти лекцій

для студентів електротехнічних спеціальностей

Російською мовою

Відповідальний за випуск *В.І Мілих*

Роботу до видання рекомендував *В.Т. Долбня*

Редактор *В.М. Баранов*

План 2007 р., поз. 95/ 199-07

Підп. до друку .28.12.2007 р. Формат 60х84 1/16 Папір офсетний.

Riso-друк. Гарнітура Таймс. Ум. друк. арк. 8.1. Обл.-вид. арк. 9,7.

Наклад 100 прим. Зам. №50. Ціна договірна

Видавничий центр НТУ «ХП»

Свідоцтво про державну реєстрацію ДК № 116 от 10.07.2000 р.

61002, Харків, вул. Фрунзе, 21

Друкарня НТУ «ХП». 61002, Харків, вул. Фрунзе, 21